Cléuzio Fonseca Filho

História da ComputaçãO

O caminho do pensamento e da tecnologia

HISTÓRIA DA COMPUTAÇÃO:
O CAMINHO DO PENSAMENTO E DA TECNOLOGIA

Cléuzio Fonseca Filho

HISTÓRIA DA COMPUTAÇÃO:
O CAMINHO DO PENSAMENTO E DA TECNOLOGIA

PORTO ALEGRE
2007

Prefácio

Esta obra se propõe a um objetivo bastante ousado: recontar a história da computação a partir de um panorama de idéias e modelos. Vence este desafio com galhardia. Neste sentido, a escolha do nome foi feliz e adequada, dado que realmente busca um ponto de vista original, fugindo de um simples relato de fatos em ordem cronológica. Bebeu na fonte de autoridades reconhecidas como o medievalista Jacques Le Goff, um dos criadores da nova historiografia, ou de historiadores da ciência do peso de Karl Popper, Thomas Khun e Imre Lakatos. Tão boa companhia certamente inspirou o autor na concepção de um todo abrangente, atualizado e inter-relacionado.

Deixa claro, por exemplo, que aquelas geringonças desengonçadas e enormes do pós-guerra e o mais moderno e colorido equipamento atual, apresentam ainda muita coisa em comum: rigorosamente seguem o mesmo princípio de funcionamento. É verdade que os computadores continuam a ganhar poder e velocidade de forma espantosa, numa evolução sem precedentes na tecnologia. Seguem possuindo os componentes estabelecidos por Von Neumann há meio século, como também a sua idéia de programa armazenado, que executado separadamente do hardware, converteu-o em uma máquina de propósito geral. No entanto, o processamento paralelo, a engenharia de software e a evolução das comunicações que culminaram na Internet, elevaram a tecnologia a patamares jamais sonhados pelos fundadores.

Por outro lado, para desenvolver as postulações que fundamentaram tamanho avanço, os criadores primevos apoiaram-se em resultados abstratos – e outros nem tanto – de pensadores do porte de Gödel, Hilbert e Turing, para citar alguns, devidamente creditados no decurso da obra. Antes mesmo de se construir a primeira máquina baseada em relês, a estrada para sua concepção estava aplainada pela contribuição destes visionários, que propuseram soluções teóricas bem à frente de seu tempo e cuja realização parecia, então, impraticável.

Esta é a melhor contribuição do livro: demonstrar que a computação nasceu do desejo de se compreender a capacidade que tem o homem em resolver problemas de forma sistemática. Assim, a tentativa de reproduzir mecanicamente estes procedimentos, muitos deles exaustivamente repetitivos, lançou as bases para estabelecer a computação como a conhecemos hoje.

A evolução dos conceitos em informática sempre esteve intrinsecamente ligada à da matemática. Nas universidades, a computação nasceu dentro dos Departamentos de Matemática. Isto justifica o resumo da história da matemática, brilhantemente apresentado segundo uma evolução de conceitos. Evitando quebrar o ritmo e sem truncar a narrativa, dá-se ao luxo de fornecer fatos curiosos e pouco conhecidos como, só para exemplificar, a dificuldade para a aceitação dos algarismos indo-arábicos por parte dos mercadores europeus, pois alguns símbolos sendo parecidos, facilitaria a falsificação.

Todo o texto está tratado de forma leve e agradável, sem se afastar do necessário rigor. A leitura flui como em um romance. Não cansa com detalhes desnecessários. Muito ao contrário, chegamos ao final desejando mais. Os anexos são oportunos, permitindo um aprofundamento de tópicos ligados à fundamentação, inadequados se incluídos no corpo principal. Vale lembrar que o primeiro deles é uma cronologia comparada, um grande esforço

de compilação, que permite contextualizar os avanços da matemática e da computação a par de outras áreas tecnológicas.

Na história mais recente da computação, não se prende somente à evolução do hardware, que foi fundamental para o desenvolvimento da disciplina, mas incapaz de justificar tamanha difusão. Mostra o crescimento das linguagens de computação – do Assembler à Java – a distinção entre os paradigmas de programação imperativas e declarativas, os aprimoramentos na arquitetura, os avanços do sistema operacional; enfim, a cristalização da Computação como Ciência. Não esquece de abordar tópicos destacados como Inteligência Artificial, Cibernética e o delicado equilíbrio entre o homem e a tecnologia.

Enfim, trata-se de uma obra surpreendentemente abrangente, dado seu tamanho compacto. Leitura fácil e ágil, despertará interesse não só de especialistas da área como também de pessoas afastadas do mundo dos computadores. No entanto, vislumbramos ganho maior para este texto no ensino de Computação, pois como afirma com propriedade na conclusão, "Cada conceito tem o seu lugar, a sua importância e a sua história que é necessário ser ensinado." Esta perspectiva sem dúvida enriquecerá a visão dos estudantes, embasando mais fortemente a essência do assunto, dando subsídos para se tornarem profissionais melhores.

Roberto Lins de Carvalho

Do autor

Fascinante! Ainda recordo esta palavra, dita por quem depois orientaria a minha dissertação de mestrado, origem deste livro: somente iria para a frente na futura tese se estivesse fascinado pelo assunto. E devo dizer que foi exatamente isso que aconteceu.

Excetuando-se alguns círculos mais teóricos, normalmente é considerado pela maioria das pessoas, inclusive dentro da própria Computação, que os dispositivos computacionais, que hoje fazem parte do nosso cotidiano, surgiram, por volta da década de 1940. O século XX teve a glória de materializar tantos artefatos, em tantas áreas, que esquecemos que na verdade são resultado, fruto, do labor de muitos que nos precederam. A Computação não escapa a essa lei. Nomes como Turing, Hilbert, Church, Frege e tantos outros até chegar a Aristóteles e aos babilônios de 4.000 a.C. misturam-se com lógica matemática, sistemas axiomáticos, formalismo e álgebra. Ao se estudar um pouco, percebe-se que toda essa tecnologia é fruto de um devir de séculos, uma auto-estrada de quase 2000 anos, paciente e laboriosamente pavimentada por figuras que são desconhecidas por muitos de nós, profissionais de informática, ou só superficialmente conhecidas.

Procurar resgatar este 'lado humano' e teórico da computação, contribuir de alguma forma para que outras pessoas da área ou de fora dela possam apreciar desde outro ângulo os alicerces deste imponente edifício formado pela tecnologia dos computadores, entusiasmar aqueles que estão entrando na área de informática, são os objetivos principais deste livro. Não pensei nada de novo, nem tive pretensões de originalidade. Afinal a história já foi feita! Tudo que escrevi já estava registrado. Apenas percebi que faltavam, e ainda faltam, trabalhos em português que tratem dos conceitos e idéias que fundamentaram a Computação. Logicamente não esgotei nenhum tema, somente procurei traçar uma linha coerente da evolução destes conceitos, aprofundando um pouco mais em um caso ou outro, procurando deixar uma boa bibliografia, embora haja muitos livros que possam ser acrescentados. Espero que este trabalho sirva como ponto de partida para outros, pois há muita coisa a ser feita para iluminar e tornar mais acessíveis determinados conceitos.

Gostaria de deixar constantes alguns agradecimentos. Em primeiro lugar ao prof. Dr. Aluízio Arcela, do Departamento de Computação da UnB, orientador da minha dissertação de mestrado e quem sugeriu e acompanhou aqueles meus primeiros estudos, base desta obra. Ao prof. Dr. Nelson Gonçalves Gomes, do Departamento de Filosofia da UnB, que tanta paciência teve para esclarecer alguns conceitos lógico-matemáticos e fornecer indicações preciosas de bibliografia. E um especial agradecimento ao prof. Dr. Roberto Lins de Carvalho (PUC-RJ) pelo incentivo que deu e entusiasmo que transmitiu ao tomar conhecimento do que estava fazendo, sem o que possivelmente não teria me atrevido a escrever coisa alguma. E aos amigos e colegas que me apoiaram e ajudaram na revisão desse trabalho, e que acabaram por lhe dar uma forma mais 'amigável'.

Cléuzio Fonseca Filho

Índice

1 Introdução

A ciência normalmente é cumulativa, isto é, constroem-se instrumentos mais poderosos, efetuam-se medidas mais exatas, precisam-se melhor e ampliam-se os conceitos das teorias, e assim por diante. Embora os paradigmas possam mudar, as pesquisas normalmente evoluem com base em resultados do passado, que se constituem em fundamentos de um desenvolvimento posterior. O cientista estará mais seguro em suas pesquisas e mais preparado para novos desafios se souber como seu assunto específico evoluiu historicamente, quais as dificuldades maiores, as soluções encontradas e os problemas pendentes.

Nas ciências mais tradicionais – Filosofia, Matemática, Física, Biologia, etc. – existem sempre estudos de história junto a muitos outros dedicados a pensadores, inventores e conquistadores de primeira, segunda ou terceira grandeza, além de inúmeras monografias. No caso da Computação, é necessário que apareçam trabalhos para servir de base e referência aos estudantes, novos pesquisadores e aqueles interessados pelos aspectos teóricos que estão por detrás dessa tecnologia que domina o cotidiano neste fim e início de milênios.

A História da Computação está marcada por interrupções repentinas, por mudanças inesperadas e imprevistas, tornando-se difícil a visão da evolução dos computadores mediante uma mera enumeração linear de invenções-nomes-datas. O desejo de conhecer as vinculações que o trabalho de determinados homens estabeleceram no tempo vem acompanhado do impulso de compreender o peso desses atos no conjunto da História da Computação. Buscar uma compreensão dos fatos através dos acontecimentos que o precederam é um dos principais objetivos que estará presente neste estudo da História da Computação.

A computação é um corpo de conhecimentos formado por uma infra-estrutura conceitual e um edifício tecnológico onde se materializam o hardware e o software. A primeira fundamenta a segunda e a precedeu. A teoria da computação tem seu desenvolvimento próprio e independente, em boa parte, da tecnologia. Essa teoria baseia-se na definição e construção de máquinas abstratas, e no estudo do poder dessas máquinas na solução de problemas. A ênfase deste livro estará nessa dimensão teórica, procurando mostrar como os homens, através dos tempos, buscaram elaborar métodos efetivos para a solução de diversos tipos de problemas.

A preocupação constante de minimizar o esforço repetitivo e tedioso produziu o desenvolvimento de máquinas que passaram a substituir os homens em determinadas tarefas. Entre essas está o computador, que se expandiu e preencheu rapidamente os espaços modernos pelos quais circulam as pessoas. A partir do aparecimento da noção de número natural, passando pela notação aritmética e pela notação mais vinculada ao cálculo algébrico, mostra-se como apareceram regras fixas que permitiram computar com rapidez e precisão, poupando, como dizia Leibniz, o espírito e a imaginação. "Descartes acreditava no emprego sistemático do cálculo algébrico como um método poderoso e universal para resolver todos os problemas. Esta crença juntou-se à de outros e surgem as primeiras idéias sobre máquinas universais, capazes de resolver todos os problemas. Esta era uma crença de mentes poderosas que deixaram obras respeitáveis na Matemática e nas ciências em geral" **[CO98]**.

Também é intenção do presente estudo procurar compreender e estabelecer as diretrizes para uma disciplina de História da Computação, mediante a seleção das idéias, teorias e conceitos que ajudaram os homens em sua busca da automatização dos processos aritméticos e que conduziram à tecnologia dos computadores.

No Brasil ainda não existem livros que tratem do assunto História da Computação, observando-se uma lacuna cultural que países do primeiro mundo preocupam-se em preencher* já faz alguns anos. Pretende-se que este trabalho seja uma contribuição nesse sentido e um ponto de partida para novos estudos de História, pois são muitos os campos que poderão ser abertos.

1.1 Ordenação dos assuntos

O desenvolvimento deste livro estará apoiado na seguinte seqüência de capítulos:

- Uma reflexão sobre a História
- Motivações para se estudar a História da Computação
- Evolução dos conceitos
- Pré-História tecnológica
- As primeiras máquinas
- A revolução do hardware e do software
- A disseminação da cultura informática e o controle das informações
- Conclusão

Primeiramente será tratado o tema da História: constatar sua existência e necessidade, aspectos da evolução da ciência histórica e tocar particularmente o tema da História da Ciência, que se relaciona com o presente trabalho. Logo a seguir virá uma breve explanação de motivos que incentivam a aprofundar no estudo do tema específico da História da Computação.

Em *Evolução dos conceitos* será mostrado o desenvolvimento dos conceitos teóricos que formaram a base para o surgimento da Computação. O caminho a ser usado será o da História da Matemática, desde os seus primórdios por volta do ano 4.200 a.C. – época provável de um calendário solar egípcio **[Boy74]** –, passando pelas contribuições das culturas babilônica, hindu, chinesa, árabe e grega, pelo ábaco, pela primeira máquina de calcular, até Boole, Hilbert, Turing e von Neumann, entre outros, nos anos 30, 40 e 50 do século XX. A partir daí, a Computação constrói a sua própria história, embora os laços com a matemática continuem sempre muito estreitos.

Por *Pré-História tecnológica* entende-se a enumeração de alguns dispositivos analógicos primitivos, as primeiras tentativas de se construir um dispositivo de cálculo com Leibniz, Pascal, Babbage, Hollerith, etc., o surgimento dos dispositivos analógicos modernos – planímetros, analisadores harmônicos, etc. – e os primeiros 'computadores' eletromecânicos por volta dos anos de 1930 e 1940.

Em *As primeiras máquinas* ver-se-á a construção dos primeiros dispositivos computacionais e os primeiros passos que são dados nesse campo essencial da Computação

* Nestes últimos dez anos vários livros já foram publicados em outros idiomas.

que são as Linguagens de Programação. Já estava formada a infra-estrutura conceitual necessária e a tecnologia já possibilitava o desenvolvimento de dispositivos mais poderosos e precisos para a execução de cálculos.

Sob o título de *A revolução do hardware e do software* abordar-se-á o desenvolvimento posterior da Computação, os avanços da Inteligência Artificial, das Linguagens de Programação e Arquitetura de Computadores. Segue-se também uma análise da Computação como uma Ciência, da Teoria da Computação, das bases matemáticas para Análise de Algoritmos, e do surgimento do tema da Complexidade Computacional.

No capítulo *A disseminação da cultura informática e a proliferação das informações* dois assuntos serão colocados. O primeiro tratará do impacto social do desenvolvimento da Computação e da necessidade de uma *análise* mais cuidadosa dos dados que os computadores tornaram disponíveis ao homem. O segundo fará algumas considerações sobre alguns limites do uso dos computadores.

2 Uma reflexão sobre a História

Uma curiosidade de explicar e compreender o mundo é o estímulo que leva os homens a estudarem o seu passado.

Arnold Toynbee

Da curiosidade do homem por si mesmo nasce a história.

A. Brunner

Na língua latina a palavra *história* expressa dois conceitos distintos: plenitude de suceder e o conhecimento que se possui desse suceder. Sua origem procede de certa raiz grega que significa inquirir, com inclinação à curiosidade **[Fer85]**.

Plenitude de suceder, conhecimento desse suceder, recuperação dos valores antigos..., palavras que significam algo mais que uma mera enumeração de nomes, lugares, datas, números, etc. Consiste antes de tudo em um debruçar-se sobre o passado e formular-lhe perguntas para se apropriar do seu legado (da "tradição", de *traducere*, entregar).

Ninguém produz por si mesmo os conhecimentos de que necessita para sobreviver em meio à sociedade na qual nasce; a grande maioria chega como algo adquirido, que se recebe pela interação com o meio ambiente. Desde o instante em que o homem se dá conta do mundo e de si mesmo, percebe-se rodeado de instituições e tradições que vive e atualiza de um modo natural, sem se dar conta de que foi forjado nesse entorno, com atitudes e pontos de vista tão arraigados em seu modo de ser, em sua psicologia, que nada lhe parece estranho ou desconhecido. Somente quando o homem sai do seu entorno vital e entra em contato com novas superfícies de valores, tradições, costumes, é que começa a compará-los com os seus e a se perguntar reflexivamente sobre tais coisas, pelas verdades de umas e outras.

A história é parte dessa necessidade humana de refletir: é o desejo de explicar a origem e a verdade das próprias instituições, quem ou qual acontecimento as estabeleceu. Para responder sobre sua existência atual e conhecer a si mesmo o homem tem de mergulhar no seu passado, perguntando às gerações anteriores por que fizeram essas instituições e não outras, por que surgiram esses precisos costumes e atitudes, por que ele tem essa herança cultural, e assim por diante. Por possuir uma herança é que cada homem é um historiador em potencial. Assim como em cada homem há uma evolução biológica necessária, há também a manutenção de uma identidade ao longo das várias etapas desse desenvolvimento biológico, que nos distinguem e nos tornam únicos, sendo fator de compreensão do modo pessoal de ser. Com a história buscamos essa nossa identidade para compreender o momento presente. E isto pode e deve ocorrer sob pontos de vista específicos: sociais, psicológicos, filosóficos e tecnológicos.

Paul M. Veyne fala ainda da história como *compreensão*, contrapondo o uso deste termo ao uso do termo *explicação*. Em seu sentido mais forte explicar significa "atribuir um fato a seu princípio ou uma teoria a outra mais geral" como fazem as ciências ou a filosofia. Nesse caso, a história seria uma difícil conquista porque a ciência só conhece leis, sistemas hipotético-dedutivos, e no mundo da história reinam, lado a lado, a liberdade e o acaso, causas e fins, etc. Para Veyne a história apresenta um caráter acientífico no sentido de que é difícil buscar princípios universais que tornem os acontecimentos inteligíveis, ou achar mecanismos de

causa e efeito para se poder deduzir, prever. "(...) a Revolução Francesa se explica pela subida de uma burguesia capitalista: isto significa, simplesmente, (...) que a narração da revolução mostra como essa classe ou seus representantes tomaram as rédeas do estado: a explicação da revolução é o resumo desta e nada mais. Quando solicitamos uma explicação para a Revolução Francesa, não desejamos uma teoria da revolução em geral, da qual se deduziria a de 1789, nem um esclarecimento do conceito de revolução, mas uma análise dos antecedentes responsáveis pela explosão desse conflito (...)". Busca-se portanto uma compreensão dos fatos através dos acontecimentos que o precederam* **[Vey82]**. Toda verdadeira investigação edifica-se estabelecendo-se com a máxima exatidão possível o já sabido, para depois poder perguntar com exatidão, de maneira que se possam encontrar respostas. Só partindo da informação adquirida podem ser feitas perguntas capazes de ter resposta, e não perguntas deslocadas, no vazio, que nunca poderão ser respondidas. É necessário caminhar passo a passo, um após o outro: em toda busca que se queira chegar a algo é preciso estabelecer com precisão o problema, planejar possíveis linhas de ataque conceitual e valorar as aparentes soluções.

Tal enfoque será um dos que estarão presentes neste estudo crítico da História da Computação através de uma visão conceitual. Pode-se aplicar a essa história a mesma afirmação que faz Thomas Khun sobre a História da Ciência: está marcada por interrupções repentinas, por inesperadas e imprevistas mudanças, exigindo modelos de conhecimento que supoêm alterações inesperadas no processo do seu desenvolvimento **([RA91], vol III)**. Em função desse fato torna-se difícil a visão da evolução dos computadores mediante uma mera enumeração linear de invenções-nomes-datas†, forçando-nos a tentar compreender as forças e tendências que, no passado, prepararam o presente. O desejo de conhecer as vinculações que os atos de determinados homens estabeleceram no tempo vai acompanhado do impulso de compreender o significado de tais atos no conjunto da História da Computação.

2.1 A História e suas interpretações

Desde o seu nascimento nas civilizações ocidentais, tradicionalmente situado na antigüidade grega (Heródoto, século V a.C. é considerado por alguns como o "pai da história"), a ciência histórica se define em relação a uma realidade que não é construída nem observada, como na matemática ou nas ciências da natureza, mas sobre a qual se "indaga", se "testemunha". Este aspecto da história-relato, da história-testemunho, jamais deixou de estar presente no desenvolvimento da ciência histórica.

---

* Se a história é ou não ciência é uma questão muito disputada entre vários autores e tema ainda polêmico. No tratado *História e Memória* do medievalista francês Jacques Le Goff **[Gof94]**, capítulo História, item 1, desenvolve-se uma panorâmica geral dessas correntes e tendências existentes entre historiadores e teóricos da história.

† Obviamente não se quer tirar aqui a importância da datação. Como diz Le Goff, "o historiador deve respeitar o tempo que, de diversas formas, é condição da história e que deve fazer corresponder os seus quadros de explicação cronológica à duração do vivido. Datar é, e sempre será, uma das tarefas fundamentais do historiador, mas deve-se fazer acompanhar de outra manipulação necessária da duração – a periodização – para que a datação se torne historicamente pensável" **[Gof94]**. Não se dispensará este trabalho de ter uma cronologia, a partir da qual se possa situar no tempo os homens e os fatos mais representativos de uma determinada corrente de idéias ou descobertas. No **anexo I** há uma tabela da evolução conceitual e tecnológica por data.

A partir do momento em que se começaram a reunir documentos escritos, a historiografia começa a ultrapassar os limites do próprio século abrangido pelos historiadores, superando também as limitações impostas pela transmissão oral do passado. Com a construção de bibliotecas e a criação de arquivos iniciou-se o desenvolvimento de métodos de crítica. Sobretudo depois do final do século XVII, estabeleceram-se os fundamentos para uma metodologia aplicada à história, sob uma radical exigência de submeter todas as investigações à razão crítica **[Fer85]**. A segunda metade do século XIX impôs o paradigma de uma história que a partir daí chamar-se-á paradigma "tradicional" ou paradigma "rankeano", derivado do nome do historiador Leopold von Ranke (1795-1886). Ranke propunha apresentar os fatos tais como o "foram na realidade" e os historiadores europeus criaram os grandes esquemas políticos e institucionais.

Características desse paradigma, conforme Peter Burke, historiador de Cambridge **[Bur92b]**:

- a história diz respeito essencialmente à política;
- é essencialmente uma narrativa de acontecimentos;
- "visão de cima" no sentido de estar concentrada nos feitos dos grandes homens;
- baseada em documentos;
- deveria perguntar mais pelas motivações individuais do que pelos movimentos coletivos, tendências e acontecimentos;
- a história é objetiva, entendendo-se por isso a consideração do suceder como algo externo ao historiador, suscetível de ser conhecido como objeto que se põe diante do microscópio, almejando uma neutralidade.

Ainda no século XIX algumas vozes soaram discordantes desse paradigma histórico. Entre outras coisas devido ao seu caráter reducionista, onde situações históricas complexas são vistas como mero jogo de poder entre grandes homens (ou países), e também em função daquilo que se poderia chamar a "tirania" do fato ou do documento, importantes sem dúvida, mas que não deve levar a abdicar de outros tipos de evidências. Como relata Peter Burke, "Michelet e Burckhardt, que escreveram suas histórias sobre o Renascimento mais ou menos na mesma época, 1865 e 1860 respectivamente, tinham uma visão mais ampla do que os seguidores de Ranke. Burckhardt interpretava a história como um corpo onde interagem três forças – Estado, Religião e Cultura – enquanto Michelet defendia o que hoje poderíamos descrever como uma 'história da perspectiva das classes subalternas'(...)" **[Bur92a]**. Outros opositores da "história política" foram os historiadores da evolução das sociedades sob o ponto de vista econômico e os fundadores da nova disciplina da sociologia, que começaram a surgir na França.

Dois fatos, no entanto, ocorridos nas quatro primeiras décadas do século XX acabariam por sacudir e arruinar a confiança nos princípios rankeanos, O primeiro foi a rápida difusão do marxismo, que renuncia à neutralidade, afirmando que o materialismo dialético é a única filosofia científica válida para a interpretação da história; o segundo, a grande crise do ano de 1929, que revelou até que ponto os fatores econômicos e sociais podem exercer uma ação decisiva.

É desse período o nascimento da revista francesa *Annales*, considerada uma das mais importantes propulsoras da chamada *Nova História*. Nova História é um termo que data de 1912, quando o estudioso americano James Harvey Robinson publicou um livro com esse título. Segundo Robinson, história inclui todo traço e vestígio de tudo o que o homem fez ou pensou desde seu primeiro aparecimento sobre a terra. Em relação ao método, a 'nova história' vai servir-se de todas aquelas descobertas que são feitas sobre a humanidade, pelos antropólogos, economistas, psicólogos e sociólogos **[Bur92b]**.

Surgiu a idéia de uma *história total*, com a qual quiseram os autores da Escola dos Annales advertir que, frente à unilateralidade e reducionismo do materialismo dialético, a compreensão do passado exige que todos os dados – políticos e institucionais, ideológicos, econômicos, sociais, da mentalidade humana, etc. – fossem fundidos e integrados para conseguir uma explicação correta. Uma tarefa árdua, na prática quase impossível, mas que marca um ideal, uma direção, uma meta que é preciso atingir.

Surgiram ainda outros enfoques como, por exemplo, a história do ponto de vista *quantitativo*, durante certo tempo em moda na Europa e Estados Unidos, que procura utilizar fontes quantitativas, métodos de contagem e até modelos matemáticos na sua pesquisa histórica, ou as histórias que abrangem um determinado campo da vida humana como a história da arte ou a história das ciências **[GN88]**.

O panorama atual, de acordo com os historiadores, é o de uma história fragmentada, detectando-se alguns sinais de busca de uma síntese. Ainda se está a uma longa distância da "história total". Na verdade, é difícil acreditar que esse objetivo possa ser facilmente alcançado – ou até que será alcançado –, mas alguns passos já foram e estão sendo dados em sua direção.

Paralelamente a todos esses esforços, surgiram também os teóricos da história, que se esforçaram ao longo dos séculos para introduzir grandes princípios que pudessem fornecer linhas gerais de compreensão para a evolução histórica. A filosofia da história é o estudo da realidade "latente", ou melhor, do "pano de fundo" dos fatos históricos. Qual é a natureza, por exemplo, das crises de crescimento e decadência de uma civilização, quais foram as causas? Sendo a história não a simples crônica que apresenta os fatos de um modo minucioso, mas sim sua investigação, que se esforça por compreender os eventos, captar relações, selecionar fatos, como fazer isso, qual é a estrutura essencial da realidade histórica?

A filosofia da história – termo temido por muitos autores porque poderia supor apriorismos, preconceitos, idealismos – responderá basicamente a duas questões fundamentais:

- o que são os fatos históricos – *historiologia morfológica*;
- para qual fim se dirigem – *historiologia teleológica.*

2.2 A História da Ciência

O nascimento e o desenvolvimento da ciência experimental, a partir do século XVII, estiveram freqüentemente acompanhados de polêmicas filosóficas: sobre o alcance do raciocínio científico, seus limites, o que é a verdade na ciência, etc. Diferentes posturas filosóficas da época moderna tentaram solucionar tais polêmicas, mas foi no século XX que

realmente se chegou a constituir uma filosofia da ciência como disciplina autônoma. Do Círculo de Viena em 1929, passando por Karl Popper, Thomas Khun, Imre Lakatos, Paul Feyerabend, Wolfang Stegmüller, entre outros, protagonizou-se um intenso debate em torno do valor do conhecimento*.

Para este trabalho, o que interessa é que toda essa movimentação em torno da racionalidade da ciência também teve seu reflexo na teoria da história, pelas novas epistemologias científicas que foram surgindo. Os debates trouxeram para o primeiro plano a questão da função da historiografia da ciência e alguns problemas teóricos relativos a essa historiografia. A importância de uma história da ciência que vá além da história episódica ou dos resultados obtidos ficou ressaltada. Em **([RA91], volume III)** resume-se quais seriam as funções da História da Ciência:

- sendo a ciência fator de história, não se pode entender o desenvolvimento dessa história, especialmente da época moderna e da época contemporânea, se não conhecermos a História da Ciência e da Tecnologia;
- além de ser fator de história, a ciência também é fator de cultura: assim, estará vedada a compreensão do desenvolvimento da cultura mais ampla se não se compreende a História da Ciência e seu entrelaçamento e condicionamento recíproco com a História da Filosofia, as concepções morais, políticas e outras;
- o conhecimento da História da Ciência é necessário para o trabalho do cientista, porque o pleno entendimento do conteúdo de uma teoria pode ser obtido mediante o confronto dessa teoria com outras, e essas outras teorias devem ser buscadas onde quer que estejam disponíveis, tanto no presente como no passado;
- a História da Ciência se revela como mais um ingrediente para a didática das ciências, tanto no que se refere à motivação do aprendizado, como no que se refere à educação no antidogmatismo, isto é, no reconhecimento do erro como uma fonte científica de aperfeiçoamento da teoria;
- a História da Ciência possibilita uma maior consciência das normas metodológicas necessárias ao trabalho de pesquisa.

Os problemas do como realizar essas funções são complexos, bastando lembrar as diferentes escolas de história. De qualquer maneira, é a disciplina da história que é revitalizada, despertando a capacidade do homem de assumir o seu passado e a partir dele dar respostas criadoras aos novos problemas que aparecem. É muito significativo que entre os sintomas da decadência de uma cultura ou de uma ciência esteja precisamente isto: o repúdio ao passado que as valorizava.

* Para aprofundar no assunto, em **[Art94]** há uma síntese das discussões e evolução das polêmicas.

2.3 Enfoque histórico adotado

A história não é exclusivamente caos ou acaso: existe no comportamento humano um certo grau de ordem e padrão observáveis de uma regularidade parcialmente previsível.

The Social Sciences in Historical Study (vários autores)

Uma das intenções do presente estudo é procurar compreender e estabelecer as diretrizes para uma disciplina da História da Ciência, a História da Computação, através da seleção das idéias, teorias e paradigmas que ajudaram os homens em sua busca da automatização dos processos aritméticos e que conduziram à tecnologia dos computadores. Interessa portanto o enfoque teleológico, citado anteriormente (*A história e suas interpretações).*

A historiologia teleológica aplica-se na interpretação, de trás para frente, da conexão concreta do curso histórico. Trata-se de compreender duas coisas: a primeira, que na série confusa dos fatos históricos podem-se descobrir linhas, facções, traços, em suma, uma 'fisionomia', conforme diz Ortega y Gasset; a segunda, tentar mostrar um sentido para a história, desde a perspectiva do seu fim. Conforme outro historiador, Toynbee* **[Toy87]**, o ponto de partida da interpretação histórica, como o de qualquer tarefa intelectual, é o pressuposto de que a realidade tem algum significado que nos é acessível pelo processo mental da explicação. Considera-se que a realidade, ainda que não totalmente, tem um sentido, isto é, que há um 'acúmulo de ordem' nas relações entre os milhares de fenômenos observados na realidade e dissecados pela nossa inteligência. "Todo raciocínio pressupõe a existência de *conexões* na natureza, ... e seu único objetivo é determinar que elementos essas conexões reúnem" **[H30]**.

Na abordagem teleológica, a história não é o fato meramente enumerado, mas organizado, selecionado, relacionado. Como diz Kenneth O. May, "de modo semelhante à física, nós não pensamos que o mero registro de uma observação por um físico é Física. Isso se torna Física quando é interpretado, organizado, relacionado com outras partes da Física. Do mesmo modo, o conhecimento cronológico torna-se história somente quando ele é selecionado, analisado, acompanhado da sua compreensão dentro de um contexto mais amplo. Significa que a história dos computadores deveria ser compreendida não do ponto de vista 'histórico', mas em relação ao computador propriamente. Deveria dar a perspectiva, através das idéias, sobre o que o futuro desenvolvimento deveria ser, ao que as futuras linhas de desenvolvimento devem chegar, e assim por diante" **[May80]**.

O conhecimento histórico, por sua própria natureza, é inseparável do historiador, pois é este que, da documentação coletada, destaca o singular, elevando o fato à condição de histórico. Procurou-se, então, registrar neste livro, tecendo um fio de história, os *fatos conceituais*, com a mínima periodização e datações possíveis. Por fatos conceituais entendam-se aqui as idéias e conceitos relevantes que fundamentaram a incansável busca pela mecanização do raciocínio. Entre estes estarão: Álgebra, Sistema Axiomático, Lógica Matemática, Sistema Axiomático Formal, Computabilidade, Máquina de Turing, Tese de Church, Inteligência Artificial, e outros mais.

---

* Se bem que em outro contexto, pois tinha uma outra linha de pensamento historiográfico, mas que serve também para o enfoque adotado no trabalho.

É uma história que se vai tornando incrivelmente complexa, conforme vai avançando no tempo. Os trabalhos isolados dos precursores da Física e da Matemática, e mais recentemente da própria Ciência da Computação, justamente por causa de seu isolamento, são relativamente fáceis de discernir. Mas a partir de 1950, com a proliferação das pesquisas nas universidades, nos grandes laboratórios, nas indústrias – privadas ou estatais –, observou-se um desenvolvimento acelerado da informática. A Ciência da Computação avançou em extensão e profundidade, tornando-se difícil até a tarefa de enumeração dos fatos. Surge a tentação de particularizar mais ainda. Pode-se falar por exemplo de uma história dos microcomputadores, tomando o ano de 1947 quando três cientistas do Laboratório da Bell Telefonia, W. Shockley, W. Brattain e J. Bardeen, desenvolveram sua nova invenção sobre o que seria um protótipo do transistor e de como, a partir daí, ano após ano, hardware e software progrediram e criaram novos conceitos, estruturas, em ritmo vertiginoso. E assim também no desenvolvimento das Linguagens de Programação, dos Compiladores, da Teoria da Computação, da Computação Gráfica, da Inteligência Artificial, da Robótica, e outras áreas. Começa a tornar-se difícil separar o que é significativo dentro do enfoque crítico adotado.

Surge o problema da delimitação das fronteiras, pois as várias especialidades se misturam muitas vezes[*], apesar de ter um corpo central definido. Para se atender à finalidade de uma História da Computação de caráter conceitual, este trabalho estará limitado prioritariamente ao campo das idéias, acenando para outros campos quando necessário se sua repercussão atingir a linha de evolução seguida.

De qualquer modo, embora enfatizando o aspecto do pensamento – o que se tinha em mente quando algo foi feito ou definido, e o que este algo fundamentará mais tarde –, será necessário o estabelecimento de alguns marcos temporais. Os acontecimentos da história produzem-se em determinados lugares e tempos. Esta pontualização possibilitará ir unificando esse suceder histórico específico de que se está tratando, em um processo único que mostre claramente a mudança, o desenvolvimento e o progresso. Não se dispensará absolutamente o uso das datas assim como dos fatos tecnológicos que possam ser considerados verdadeiras mudanças de paradigma[†].

Não se deve estranhar o recurso à História da Matemática. Aliás é preciso dizer que, no início, pelo menos nos círculos acadêmicos, a Computação apareceu como algo dentro dos Departamentos de Matemática, e ainda hoje, em muitas Universidades, a Ciência da Computação aparece como um Departamento de um Instituto de Matemática. Dentre os diversos tópicos científicos sujeitos à investigação, a Matemática é o que melhor combina um caráter abstrato com um uso universal em outros campos do conhecimento. Sua relação com a Computação é muito estreita, quase que inseparável. As primeiras máquinas construídas foram resultado de buscas por parte dos membros dessa comunidade do conhecimento.

---

[*] Pense-se na Robótica por exemplo, onde estão incluídas a Inteligência Artificial, as Linguagens de Programação, a Computação Gráfica, etc.

[†] No anexo I encontra-se uma tabela cronológica dos acontecimentos *conceituais e tecnológicos*, que dará uma visão mais geral da evolução da História da Computação.

3 Motivações para se estudar a História da Computação

Uma vez apontada a importância e necessidade do estudo da história em geral e, mais especificamente, da história da ciência e da tecnologia, fica fácil perceber que o estudo da História da Computação é um interessante relevo dentro da vasta paisagem do conhecimento científico. Basta lembrar que o impacto dessa tecnologia na nossa sociedade é imenso e nossa dependência dela cada vez maior.

Seguem abaixo outros fatores motivadores para esse estudo.

3.1 Necessidade de discernir fundamentos

Comparada com outras áreas, a Ciência da Computação é muito recente. Mas, nestes poucos anos (pode-se apontar a Segunda Guerra Mundial como um marco inicial, quando efetivamente se construíram os primeiros computadores digitais) o avanço da Computação foi exponencial, abrindo-se em um grande leque de tecnologias, conceitos, idéias, transformando-se em uma figura quase irreconhecível. Atualmente falar de estado da arte na Computação tornou-se sem sentido: sob que ótica, perspectiva, campo ou área? Apesar da sua recente irrupção na história contemporânea, a partir dos anos 40 do século XX, ela já se tornou complexa, ampla, geradora de novos enfoques, tornando-se um verdadeiro desafio a quem queira entendê-la e traçar sua evolução.

Ao mesmo tempo, cada nova geração de informatas depara-se com um duplo problema: a impossibilidade de ter uma visão global sobre todo o conhecimento precedente e, mais acentuadamente ainda, a história do desenvolvimento das várias especialidades. Não estão individualizados os eventos, por vezes complexos, que antecederam o saber atual e também não se possui um quadro que os reúna, para se ter uma idéia geral, coerente e significativa. A evolução tecnológica se nos apresenta abrupta, através de saltos descontínuos, e todo o trabalho que antecede cada etapa aparece coberto por uma camada impenetrável de obsolescência, algo para a paleontologia ou para os museus, como se nada pudesse ser aprendido do passado.

O resultado é um empobrecimento do panorama atual da realidade da informática. Não se estabelecem conexões entre os vários campos da Ciência da Computação, caindo-se facilmente no utilitarismo. As camadas mais profundas dos conceitos não são atingidas, o conhecimento torna-se bidimensional, curto, sem profundidade.

Junto a isso, cedendo talvez a um imediatismo ou deixando-se levar por uma mentalidade excessivamente pragmática de busca de resultados, há uma forte tentação de se estabelecerem ementas para o estudo da Ciência da Computação preocupando-se mais com determinados produtos – linguagens, bancos de dados, sistemas, aplicativos, etc. – e pouco se insiste na fundamentação teórica.

Os matemáticos aprendem aritmética e teoria dos números, pré-requisitos sem os quais não se evolui no seu campo do saber; os engenheiros, cálculo diferencial, física; os físicos trabalham arduamente na matemática, e assim por diante. Quais os fundamentos correspondentes na Computação? Conhece-se a *Álgebra Lógica* de George Boole, um matemático que buscando relacionar o processo humano de raciocínio e a Lógica Matemática,

desenvolveu uma ferramenta para os futuros projetistas de computadores? Sabe-se que a revolução da Computação começou efetivamente quando um jovem de 24 anos, chamado Alan Mathison Turing, teve a idéia de um dispositivo teórico para buscar a resposta a um desafio do famoso matemático David Hilbert – um dos primeiros a falar sobre computabilidade –, e que em um 'journal' de matemática comentou aos seus colegas que era possível computar na teoria dos números, por meio de uma máquina que teria armazenadas as regras de um sistema formal? Que as pesquisas de Turing estão relacionadas com o trabalho de Gödel – cujo Teorema que leva o seu nome é considerado um dos mais famosos resultados do século XX, dentro da matemática? Pode-se citar ainda a Tese de Turing-Church que possibilitou aos cientistas passarem de uma idéia vaga e intuitiva de *procedimento efetivo* para uma noção matemática bem definida e precisa do que seja um algoritmo. E antes de todos esses, o esforço de dezenas de pensadores de diferentes culturas, para encontrar melhores formas de usar símbolos, que viabilizou o desenvolvimento da Ciência Matemática e Lógica, e que acabaram fundamentando toda a Computação.

3.2 Incentivo à educação para a qualidade do software

A investigação histórica sob o prisma citado – das idéias e conceitos fundamentais que formaram a base do desenvolvimento da Computação – poderá contribuir para uma questão que assume importância decisiva e crucial: a qualidade do software.

É preciso aqui tecer um comentário relacionado ao tema da qualidade. A expressão "crise do software" **[Nau69]** apareceu no final da década de sessenta na indústria tecnológica da informação. Referia-se aos altos custos na manutenção de sistemas computacionais, aos custos relativos a novos projetos que falhavam ou que consumiam mais recursos que os previstos, etc., realidades presentes no dia a dia de muitos centros de processamento de dados. Ao lado disso, havia, e ainda há, uma disseminação anárquica da cultura informática, impregnando cada dia mais a vida social e trazendo, como conseqüência, uma dependência cada vez maior da sociedade em relação ao computador. Torna-se fundamental, portanto, diminuir as incertezas presentes no processo de elaboração dos sistemas de computação.

A resposta a esses desafios já há alguns anos vem sendo formulada no sentido de se estabelecer uma execução disciplinada das várias fases do desenvolvimento de um sistema computacional. A *Engenharia de Software* surgiu tentando melhorar esta situação, propondo abordagens padronizadas para esse desenvolvimento. Algumas dessas propostas vão em direção ao uso de métodos formais* nos processos de elaboração do sistema, basicamente através da produção de uma especificação formal em função das manifestações do problema do mundo real que estiver sendo tratado e através da transformação dessa especificação formal em um conjunto de programas executáveis.

Há também pesquisas dentro da Computação que caminham em direção ao desenvolvimento de métodos numéricos, em direção à lógica algébrica. George Boole, em sua obra que deu início a uma nova arrancada no desenvolvimento da Lógica Matemática e da

* Um método se diz formal quando o conjunto dos procedimentos e técnicas utilizadas são formais, isto é, têm um sentido matemático preciso, sobre o qual se pode raciocinar logicamente, obtendo-se completeza, consistência, precisão, corretude, concisão, legibilidade e reutilização das definições abstratas.

Computação, como se verá, já dizia: "A lógica simbólica ou lógica matemática nasce com a vocação de ferramenta para inferência mecanizada através de uma linguagem "simbólica"**[Boo84].** Busca-se por essa vertente a criação de uma metalinguagem lógico-matemática para o desenvolvimento de sistemas, de tal maneira que se possa constituir no instrumento que transfere a precisão da matemática aos sistemas. A figura abaixo, conforme **[Coe95]**, serve como ilustração dessa idéia.

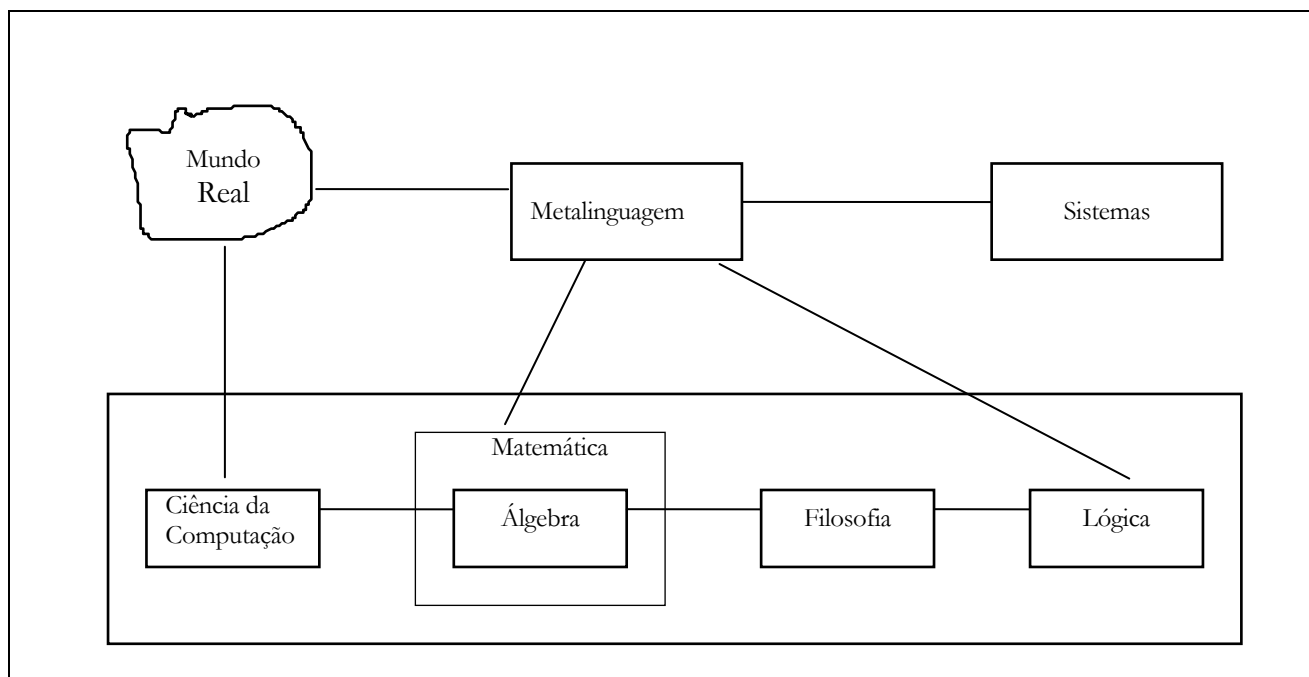



Figura 1: O desenvolvimento de sistemas através de especificações formais

Técnicas provenientes da área da lógica matemática vêm sendo aplicadas a diversos aspectos do processo de programação de computadores*. Não é sem dificuldades que evoluem as investigações sobre o papel do raciocínio formal matemático no desenvolvimento do software. Mas é preciso ressaltar a sua importância pois através da especificação formal e verificação obtém-se uma excelente guia para o desenvolvimento de programas corretos e passíveis de manutenção. Talvez não se possa pedir que todo programa seja formalmente especificado e verificado, mas é de se desejar que todo programador que almeje profissionalismo esteja ao menos familiarizado com essas técnicas e seus fundamentos matemáticos **[BBF82]**.

Concluindo-se o comentário sobre os problemas relativos à qualidade do software, pode-se dizer que a solução para o problema do desenvolvimento de software passa por um

* Entre os aspectos estudados encontram-se: (i) *verificação* (prova de corretude): a prova de que um dado programa produz os resultados esperados; (ii) *terminação*: a prova de que um dado programa terminará eventualmente a sua execução; (iii) *derivação* (desenvolvimento): construção de um programa que satisfaz a um conjunto de especificações dadas; (iv) *transformação*: modificação sistemática de um programa dado para obter um programa equivalente, como estratégia para a derivação de novos programas por analogia a soluções conhecidas ou como método de otimização da eficiência de programas **[Luc82]**.

treinamento formal mais intenso na formação dos futuros cientistas da computação. Desde os primeiros anos da universidade é necessário que se estudem os princípios matemáticos, e em alguns casos até físicos (como se poderá falar em Computação Gráfica se não se sabe quais são as propriedades de um sistema de cores), que formam o substrato da Computação: o que é computável (metamatemática, computabilidade), a complexidade que exige a execução de um cálculo (análise de algoritmos, teoria da complexidade), fundamentos de algumas abstrações (teoria dos autômatos, teoria das linguagens formais, teoria de rede, semântica denotacional, algébrica, etc.), fundamentos dos raciocínios que se fazem em programação (sistemas de provas, lógica de Hoare), etc. Ou seja, seria interessante notar que na Ciência da Computação há um forte componente teórico. É um corpo de conhecimentos, sistematizado, fundamentado em idéias e modelos básicos que formam a base das técnicas de engenharia e eletrônica usadas na construção de computadores, tanto no referente ao hardware como ao software. Como não falar de indução matemática quando se deseja seriamente explicar a programação de computadores? Ou falar de uma linguagem de programação sem introduzir a teoria dos autômatos? Com o desenvolvimento de conceitos matemáticos adequados será possível estabelecer um conjunto de procedimentos que assegure aos sistemas a serem desenvolvidos uma manutenção gerenciável, previsível e natural, como ocorre na engenharia.

O estudo da História da Computação, não já sob o enfoque de datas e nomes – importantes também e necessários, para não se cair na pura especulação –, mas sob o aspecto das idéias, de seus fundamentos e suas conseqüências, pode ser uma sólida base, um ponto de partida, para sensibilizar e entusiasmar o aluno sobre a importância dos fundamentos teóricos, para ajudá-lo a ver o que um determinado conceito tem como pressupostos.

3.3 Tornar claros e ligar os fatos

Entre os objetivos da ciência histórica pode-se aceitar como axiomático o de procurar dar um significado aos acontecimentos. É a busca de se dar sentido à história. Este trabalho, feito dentro de uma perspectiva teleológica da história, procurará estabelecer uma conexão causal entre eventos, para se começar a entender o sentido do passado, dispondo-o numa espécie de sistema organizado, para torná-lo acessível à compreensão.

Sob outro enfoque, pode-se ver a História da Ciência da Computação como um "olhar para trás com o fim de descobrir paralelismos e analogias com a tecnologia moderna, com o fim de proporcionar uma base para o desenvolvimento de padrões através dos quais julguemos a viabilidade e potencial para uma atividade futura ou atual" **[Lee96]**. Quer dizer, analisar o passado e reconhecer tendências que nos permitam prever algum dado futuro. Dizia John Backus, criador do FORTRAN e da Programação Funcional: "Na ciência e em todo trabalho de criação nós falhamos repetidas vezes. Normalmente para cada idéia bem sucedida há dúzias de outras que não funcionaram" (**1994, discurso ao receber o prêmio Charles Stark Draper).** A história cataloga e registra tais falhas, que então se tornam uma fonte especial de aprendizado, ensinando tanto quanto as atividades bem sucedidas. Mais ainda, dão-nos o caminho, por vezes tortuoso, por onde transcorreram as idéias, as motivações, as inovações. Ou seja, é interessante e instrutivo o estudo da história de qualquer assunto, não somente pela ajuda que nos dá para a compreensão de como as idéias nasceram

– e a participação do elemento humano nisso – mas também porque nos ajuda a apreciar a qualidade de progresso que houve.

3.4 Acompanhar novas tendências

É sintomático notar que pela primeira vez incluiu-se no curriculum para Ciência da Computação, desenvolvido pela ACM (Association for Computing Machinery) e IEEE (Institute of Electrical and Electronics Engineers) Computer Society Joint Task Force, em 1991, módulos relativos à história em 4 áreas: Inteligência Artificial, Sistemas Operacionais, Linguagens de Programação e Temas Sociais, Éticos e Profissionais. Mais recentemente ainda, na 6ª IFIP (International Federation for Information Processing), evento realizado dentro da Conferência Mundial dos Computadores na Educação, transcorrida em Birmingham, Inglaterra, de 20 a 24 de julho de 1995, estimulou-se não só "a preservação das peças de computadores, o registro de memórias dos pioneiros e a análise do impacto exterior das inovações nos computadores, mas também o desenvolvimento de módulos educacionais na História da Computação" **[Lee95]**.

Significativo também é a introdução, nos cursos de Ciência da Computação, da disciplina História da Computação, principalmente a partir da década de 1990, em algumas universidades. Pode-se citar a Universidade de Stanford e o Instituto Charles Babbage, da Universidade de Minnesota dedicado a promover o estudo da História da Computação, EUA, o arquivo Nacional para a História da Computação da Universidade de Manchester, Inglaterra, Universidade de Waterloo (Canadá) e similar em Bordeaux, França, Universidade de Wales Swansea, Austrália, etc.

Também aumentaram o números de museus e instituições governamentais ou particulares que prestam esse serviço de preservação da história da tecnologia informática, como por exemplo o museu de Boston, os museus de instituições militares americanas e organizações do porte do IEEE. Esta última promoveu em 1996 o lançamento de pelo menos quatro livros sobre o assunto História da Computação, tendo construído um "site" na Internet, narrando os eventos dessa história desde o século XVII. Na Internet proliferaram os museus de imagens e cronologias sobre assuntos específicos como Microcomputadores, Computação Paralela, Linguagens de Programação, etc.

3.5 Revalorizar o fator humano

Finalmente há o grande tributo que se deve fazer a esses homens que, ao longo da história da ciência Matemática, Lógica, Física, e mais recentemente da Computação, não se deixaram levar pelo brilho atraente daquilo que chama a atenção e das demandas mais imediatas. Motivados pela pura busca do saber formaram o arcabouço, a infra-estrutura que

possibilitou a revolução da informática. Os bits e todas as partes de um computador (incluindo o software) são na verdade o resultado de um processo, de uma evolução tecnológica de vários séculos, partilhada por inúmeros personagens, cada um acrescentando sua pequena ou grande contribuição.

"Qualquer que seja, porém, o destino da informática, ela já tem o seu lugar na História, constituindo-se num dos fatores preponderantes que moldam o conturbado mundo no fim do século XX. Sem a compreensão do seu papel social, não será possível entender o processo histórico em marcha, nem a direção do futuro. Desse modo, a pesquisa da História da Computação tem um significado fundamental no presente" **[Mot96]**.

4 Evolução dos conceitos

Considerando as idéias e os conceitos como uma das linhas que conduzirão ao grande desenvolvimento tecnológico da Computação a partir da década de 40 do século XX, este capítulo faz referência a alguns aspectos da evolução da Matemática, e mais especificamente de alguns dos seus ramos, no caso a Álgebra e a Lógica Simbólica ou Matemática, de onde nos vieram o rigor e o método axiomático, até chegar às noções de *computabilidade* e *procedimento*, com Turing e Church.

4.1 Primórdios

4.1.1 A evolução do conceito de número e da escrita numérica*

Talvez o passo mais fundamental dado nestes primeiros tempos tenha sido a compreensão do conceito de *número*, isto é, ver o número não como um meio de se contar, mas como uma idéia abstrata. O senso numérico foi o ponto de partida. Trata-se da sensação instintiva que o homem tem das quantidades, atributo participado também pelos animais irracionais (a gata mia quando um dos filhotes não está no ninho; determinados pássaros abandonam o ninho quando um dos seus ovos foi mudado de lugar); na vida primitiva bastava esse senso numérico. Mas com o começo da criação dos animais domésticos era necessário saber algo mais, pois se a quantidade não fosse melhor conhecida, muitas cabeças se perderiam. Inventou-se a contagem através do estabelecimento de uma relação entre duas ou várias quantidades, na qual cada elemento de uma corresponde a um elemento de outra e nenhum elemento deixa de ter o seu correspondente (por exemplo, pedrinhas de um monte com ovelhas de um rebanho).

No entanto essa relação biunívoca se dá somente no âmbito mental (ovelhas e pedra estão na natureza e não se dão conta um do outro). Não está registrado de que forma ocorreu o reconhecimento, pelos nossos antepassados mais primitivos, de que quatro pássaros caçados eram distintos de dois, assim como o passo nada elementar de associar o número quatro, relativo a quatro pássaros, e o número quatro, associado a quatro pedras. Essa correspondência é um pensamento que é uma espécie de linguagem. Nessa linguagem estão envolvidas a quantidade, a correspondência biunívoca (o número) e a sua expressão (os elementos usados para contagem: pedras, dedos, seqüências de toques no corpo, e outras formas mais primitivas de expressar um numeral).

A visão do número como uma qualidade de um determinado objeto é um obstáculo ao desenvolvimento de uma verdadeira compreensão do que seja um número. Somente quando, de acordo com um dos exemplos dados, o número quatro foi dissociado dos pássaros ou das pedras, tornando-se uma entidade independente de qualquer objeto – uma abstração, como diriam os filósofos –, é que se pôde dar o primeiro passo em direção a um sistema de notação, e daí à aritmética. Conforme Bertrand Russell, "foram necessários muitos

* Este item está baseado em **[Dan54]**, **[Wil97]** e **[New56]**, principalmente no segundo

anos para se descobrir que um par de faisões e um par de dias eram ambos instâncias do número dois" **[Dan54]**.

E assim como se criaram símbolos escritos para expressar idéias, também criou-se a escrita numérica. Os numerais escritos surgem nas civilizações antigas (egípcia, babilônica e chinesa) e se baseiam na repetição de símbolos. No caso dos egípcios, ao se completar o décimo elemento, tomava-se um outro símbolo para representar o número.

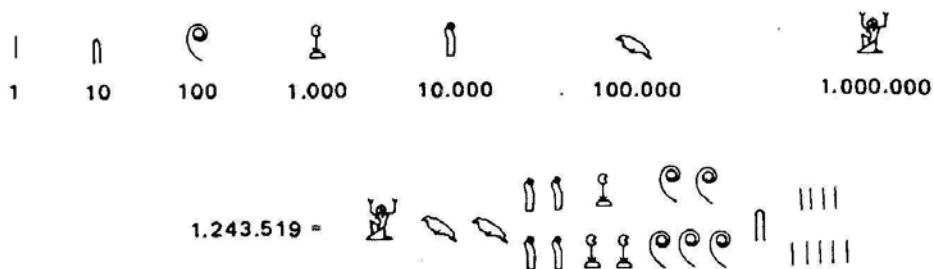



Figura 2: Sistema numérico clássico de adição egípcio baseado em hieróglifos **[Wil97]**

Foram criadas várias escritas numéricas: os gregos e os romanos usavam letras do alfabeto. Os algarismos romanos foram tão difundidos que existem até hoje, pelo menos para certas funções especiais como marcar horas em relógios e numerar capítulos em livros, especialmente livros formais como a Bíblia. Este amplo processo de criação conclui-se com a escrita numérica criada pelos hindus, há vários séculos.

O advento do que nós chamamos de sistema numérico hindu-arábico, com seu rígido esquema de valores e posições, juntamente com o zero (que era usado para representar um espaço em branco), foi uma das grandes invenções da humanidade, e possibilitou o desenvolvimento dos métodos matemáticos e aritméticos, que a partir disso evoluíram muito mais do que qualquer coisa que se conhecia até então[*].

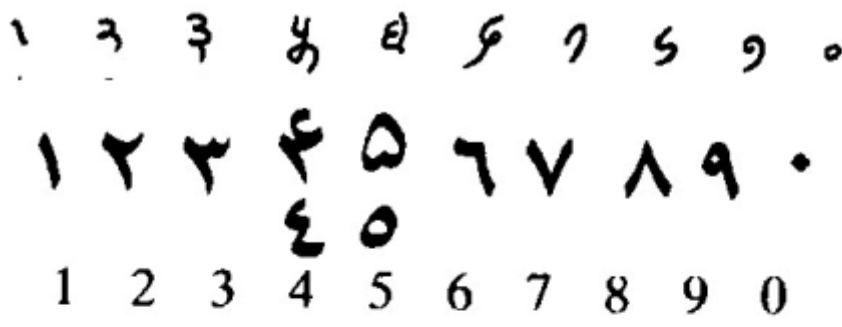
O uso do zero não era incomum em certos sistemas numéricos posicionais primitivos. Os babilônios usaram um caráter parecido com o zero para representar uma coluna vazia no meio do número por volta do ano 200 a.C., mas isso não era muito comum no sistema deles. Se um número, por exemplo 1024, precisasse de um zero para separar o dígito 1 do dígito 2, eles usavam o símbolo do zero do mesmo modo que se faz hoje. No entanto, se eles tivessem que representar um número como o 1000, parece que eram incapazes de conceber o fato de que o símbolo zero pudesse ser usado para simplesmente "cobrir" os espaços restantes, e eles representavam apenas o número 1, deixando ao leitor a tarefa de descobrir se aquilo significava 1000, e não 10, 100 ou até mesmo 1.

[*] Olhando-se para trás na História, parece que a invenção de um sistema numérico de posições rígidas, e ainda de um símbolo para designar o zero, deveria ter sido um extensão óbvia de alguns dos primeiros sistemas numéricos de posições. Essa idéia é falsa: basta pensar que escapou à percepção de grandes autores da Antigüidade, como Arquimedes ou Apollonius de Pergam, mesmo quando eles percebiam as limitações de seus sistemas.

Exatamente onde e quando os homens começaram a utilizar o atual sistema numérico posicional, e os 10 dígitos em que eles eram baseados, continua a ser um fato obscuro. Certamente veio até a Europa pelos árabes, e é bem certo de que eles o obtiveram do povo do subcontinente indiano. Onde e quando os indianos obtiveram esse sistema não é conhecido. Pode ter sido uma invenção indígena, ou ter vindo do leste da Indochina, ou um desenvolvimento do uso babilônico do símbolo da coluna *vazia*. Na Índia antiga, bem como em muitas sociedades, a arte da aritmética foi desenvolvida em um maior grau do que o necessário para o comércio, por causa da sua importância para a religião local. Todas as três primeiras religiões indianas (Janaísmo, Budismo e Hinduísmo) consideravam a aritmética importante, como mostra o fato de ser exigida entre os estudos fundamentais a serem feitos pelos candidatos ao sacerdócio.

O uso mais antigo que se tem notícia da matemática indiana está em trabalhos escritos em forma de verso, onde complicados expedientes literários eram utilizados para representar números, de modo a se preservar a rima e a métrica dos poemas. Até mesmo documentos que usam numerais para denotar números nem sempre são guias seguros para informar quando tal prática começou a aparecer. Parece que, em alguma época no século XI, foi feita uma tentativa para se racionalizar o sistema de propriedade da terra em partes da Índia, o que levou muitas pessoas a produzirem documentos forjados para pedir seus vários lotes. Das 17 inscrições conhecidas usando numerais antes do século X, todas, à exceção de duas, mostraram ser falsificações. A mais antiga e indubitável ocorrência do zero na inscrição escrita na Índia foi em 876 d.C., com os números 50 e 270 sendo representados em uma versão local dos dígitos indianos.

A história do nosso sistema numérico fica muito mais clara a partir do século IX d.C. No século VII, quando a dinastia dos Califas começou em Bagdá, o aprendizado das culturas adjacentes foi absorvido em uma nova e expansiva cultura árabe. Quando os árabes conquistavam um país, eles costumavam adquirir seu modo de escrita, particularmente a notação dos numerais do povo conquistado e procurar traços de conhecimento na literatura que sobreviveu à guerra.

Graças aos trabalhos do matemático al-Kharazmi (mais a frente se falará da importância deste homem originário da Pérsia), o uso dos numerais hindus rapidamente se expandiu por todo o império árabe. A eventual expansão desses numerais pela Europa é mais facilmente explicada a partir dos contatos gerados entre árabes e europeus pelo comércio e pelas guerras. É provável que os comerciantes italianos conhecessem o sistema de contas de seus parceiros comerciais, e que os soldados e sacerdotes que retornaram das cruzadas também tivessem uma ampla oportunidade de ter contato com o sistema de notação e aritmética árabes. O mais antigo manuscrito europeu contendo numerais hindu-arábicos de que se tem notícia foi escrito no claustro Albeda, na Espanha em 976 d.C. Os novos numerais também foram encontrados em outro manuscrito espanhol de 992 d.C., em um manuscrito do século X encontrado em St. Gall, e em um documento do Vaticano de 1077 d.C. Entretanto seu uso não foi muito difundido durante esse período inicial, e é provável que pouquíssimas pessoas tenham entendido o sistema antes da metade do século XIII.

Manuscrito espanhol datado de 976 com os nove algarismos "indo-árabes": trata-se da mais antiga menção conhecida do uso desses algarismos na Europa.

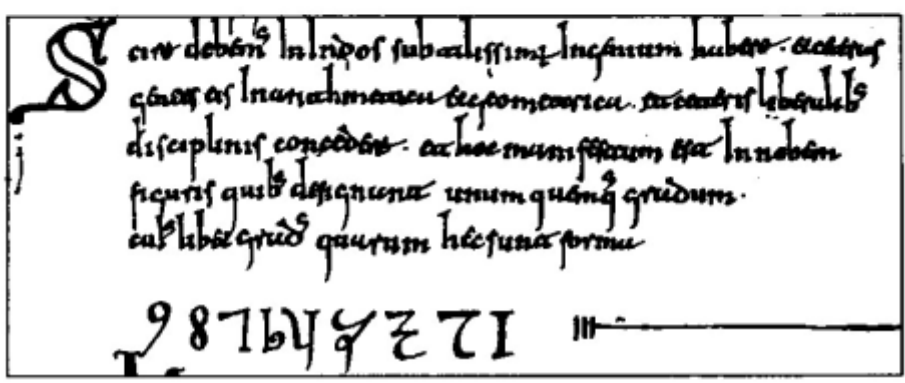



Figura 3: Mais antigo manuscrito europeu com numerais indo-arábicos, cfr. **[Ifr89]**

A primeira grande tentativa de introduzir essa nova forma de notação foi feita por Leonardo de Pisa (1175 a 1250), mais conhecido pelo nome de Fibonacci (que veio de filius Bonaccio, o filho de Bonaccio), um dos melhores matemáticos europeus da Idade Média. Durante o tempo de Fibonacci, Pisa era uma das grandes cidades comerciais da Itália, e por isso entrou em contato com toda a área do Mediterrâneo. O pai de Fibonacci era o chefe de uma das casas de comércio ultramarino, em Bugia, na costa da África Norte. Bugia era um importante centro para mercadores e estudantes da época e Fibonacci foi mandado, quando tinha 12 anos, para se juntar a seu pai, tendo uma chance de ouro para observar os métodos árabes. Certamente obteve parte de sua educação enquanto estava em Bugia, e a lenda diz que ele aprendeu árabe e aritmética por um mercador local. Visitou depois o Egito, Síria, Grécia e França, onde se esforçou para se informar sobre os sistema aritméticos locais. Ele achou todos esses sistemas numéricos tão inferiores aos que os árabes utilizavam que, quando voltou a Pisa, escreveu um livro para explicar o sistema árabe de numerais e cálculo. Esse livro, nomeado de *Liber Abaci* (O Livro do Ábaco) foi publicado pela primeira vez em 1202, e revisto e ampliado em 1228. Era um tomo muito grande para a época, constituído de 459 páginas divididas em 15 capítulos. Os capítulos 1 a 7 introduziam a notação árabe e as operações fundamentais com números inteiros; os capítulos 8 a 11 tratavam de várias aplicações, enquanto os restantes eram dedicados aos métodos de cálculo envolvendo séries,

proporções, raízes quadradas e cúbicas, e uma pequena abordagem sobre geometria e álgebra. Foi em um desses últimos capítulos em que ele introduziu o famoso
problema do coelho e as séries de números que agora levam seu nome.

O *Liber Abaci* não foi tão influente quanto deveria ser porque era muito grande, e portanto difícil de copiar em uma época em que não havia imprensa. Também continha material avançado que só poderia ser entendido por estudiosos, tendo sido conhecido apenas por poucas pessoas, nenhuma das quais parecia ter muita influência nos métodos de cálculo usado nas transações diárias.

Mas embora os esforços de Fibonacci tivessem pouco sucesso, a idéia dos numerais hindu-arábicos foi gradualmente se expandindo na Europa. As principais fontes de informação foram as várias traduções, algumas parciais, do trabalho de al-Kharazmi. O fato de ser a língua árabe totalmente diferente de qualquer língua européia foi uma grande barreira para a disseminação das idéias científicas árabes. Para aprender o árabe, era geralmente necessário viajar a um país de língua árabe, e isso era uma tarefa difícil, já que alguns árabes não eram simpáticos aos visitantes cristãos (e vice-versa). Esse problema foi parcialmente resolvido em 1085, quando Alphonso VI de Leon recapturou Toledo dos mouros e uma grande população de língua árabe veio à esfera da influência européia. A maioria das primeiras traduções, ou pelo menos as pessoas que ajudaram os tradutores, vieram dessa população.

Os dois principais trabalhos que espalharam o conhecimento aritmético hindu-árabe pela Europa foram o *Carmen de Algorismo* (o Poema do Algorismo) de Alexander De Villa Dei por volta de 1220 e o *Algorismus Vulgaris* (Algorismos Comuns), de John de Halifax, mais conhecido como Sacrobosco, por volta de 1250 d.C. Esses dois livros foram baseados, pelo menos em parte, nos trabalhos de al-Kharazmi ou de um de seus sucessores. Foram elaborados para uso em universidades européias e não pretendiam ser explicações completas do sistema; preferiram dar simplesmente o básico que o professor pudesse explicar, linha a linha, para seus alunos. O *Carmen de Algorismo* era particularmente difícil de ser seguido, especialmente na discussão do cálculo de raízes, porque foi escrito em versos hexassílabos. Apesar disso ele ficou muito popular, sendo copiado muitas vezes no latim original e sendo até traduzido em inglês, francês e irlandês. Parte dessa popularidade se deveu ao fato de que Alexander de Villa Dei (? a 1240), que era nativo da Normandia e escrevia e ensinava em Paris, já era famoso por uma gramática de latim, também em versos, que era muito utilizada nas escolas da época. Também o fato de possuir somente 284 linhas o fazia facilmente copiável pelos escribas, e assim, enquanto se produzia uma cópia do *Liber Abaci,* centenas de cópias do *Carmen* podiam ser feitas e distribuídas. O mesmo acontecia com o *Algorismus Vulgaris*, que tinha somente 4.000 linhas. Sacrobosco, que também ensinava em Paris durante a primeira metade do século XIII, era conhecido por seu trabalho em astronomia, e isso sem dúvida contribuiu para o sucesso do *Algorismus Vulgaris*, que continuou a ser usado como um texto universitário em aritmética até mesmo depois da invenção da imprensa. Edições impressas são conhecidas a partir do fim dos séculos XV e XVI.

Uma dos primeiros tradutores do trabalho de Al-Kharazmi foi Adelard de Bath que, por volta do ano 1120, produziu um texto em latim cujas primeiras palavras eram *Dixit Algorismi* … (assim disse o algorismo…), e que resultou nessa nova ciência que ficou conhecida como algorismo. Esse termo, e as várias corruptelas originadas por autores

diferentes, finalmente se espalhou através de todas linguagens européias até o ponto de o processo de fazer aritmética com os numerais hindu-arábicos ser chamado *algarismo*, e isso nos deu o termo *algoritmo* que é tão familiar aos estudantes de Ciência da Computação. A conexão do *Algorismi* com al-Kharazmi perdeu-se e muitos autores inventaram até outras pessoas, como um que citava o 'Rei Algor', a quem a origem desses métodos poderia ser atribuída...

A troca dos numerais adicionais romanos para o sistema posicional dos hindu-árabes foi lenta, durou alguns séculos. Não era fácil para os europeus entenderem o uso do zero que, enquanto representando o nada em si, podia magicamente fazer outros dígitos crescerem dramaticamente em valor. A palavra hindu para o sinal do zero era *sunya* que, muito apropriadamente, queria dizer *vazio* ou *desocupado*. Quando o sistema foi adotado pelos árabes, eles usaram sua própria palavra para desocupado, que é geralmente escrita como *sifr* no nosso alfabeto. Essa palavra árabe foi simplesmente escrita no alfabeto latino ou como *zephirum*, de onde veio a nossa palavra zero, ou como *cipher*, de onde derivou o antigo verbo inglês *to cipher*, que significava "fazer aritmética". Parte do mistério com que o novo sistema era considerado pode ser observado pelo fato de que a mesma palavra para a raiz elevou termos que eram envolvidos em mágica e escrita secreta, como *calcular* ou *decifrar* um texto em código. Esse ar de mistério foi realçado pela atitude de algumas pessoas que, depois que conheciam o algarismo, acharam que isso era um conhecimento para ser mantido entre um grupo secreto, e não para ser explicado para pessoas comuns. Essa atitude é ilustrada em várias figuras do fim da Idade Média, uma das quais mostra duas pessoas fazendo aritmética, uma usando os métodos antigos do ábaco enquanto o outro, escondendo seu trabalho do primeiro, estava usando o novo algarismo.

Por volta de 1375 o uso dos numerais hindu-arábicos firmou-se na Europa. Eles começaram a aparecer em muitos documentos diferentes, embora ainda existisse uma grande resistência para a adoção dos novos números. Em 1229, a cidade de Florença proclamou uma lei que proibia o uso dos numerais hindu-arábicos, pois eram fáceis de serem alterados ou forjados (por exemplo, transformar um 0 em um 6 ou 9 deveria ser bastante fácil). Os mercadores desenvolveram vários truques para prevenir esse tipo de coisa com os numerais romanos; por exemplo, XII era escrito como Xij, então um i extra não poderia ser adicionado ao fim sem gerar suspeitas e, fazendo o primeiro caracter maiúsculo, eles evitavam que qualquer um colocasse caracteres à esquerda do número. Passaria apenas pouco tempo para que se desenvolvessem dispositivos semelhantes para os novos numerais, mas ainda por volta de 1594 os mercantes da Antuérpia eram alertados para que não os usassem em contratos ou em ordens de pagamento bancárias.

Os italianos rapidamente viram a utilidade do novo sistema para propósitos mercantis e influenciaram toda a Europa para a adoção do sistema de valor posicional. Algarismos já eram bem difundidos por volta de 1400, mas os mercadores mais conservadores continuaram utilizando os numerais romanos até por volta de 1550, e muitos monastérios e faculdades os usaram até o meio do século XVII. Até mesmo por volta de 1681 encontram-se evidências de que o novo sistema ainda não tinha sido completamente compreendido. Um livro publicado naquele ano teve seus capítulos numerados como: 1, 2, 3, 4, 5, 6, 7, 8, 9, X, X1, X2, X3, X4 … XXX, XXX1, 302, 303… XXXX, 401, 402,…

Essa permanência dos métodos antigos de notação foi causado não por falta de conhecimento sobre o novo, mas pelo medo de que, sem um entendimento completo do sistema, alguma coisa poderia dar errado. Esse tipo de medo é visto de vez em quando até hoje, mas dois exemplos do século XVII podem ajudar a ilustrar o fenômeno. Willian Oughtred, que se encontrará mais a frente neste livro ao se falar da régua de cálculo, preferia calcular ab + ac a calcular a(b+c) por causa do medo de que algum tipo de erro poderia acontecer em um sistema abreviado. Muitas formas diferentes de numerais foram usadas na Europa, alguma das quais não seriam reconhecidas da maneira que elas são agora. As versões manuscritas dos antigos trabalhos são particularmente difíceis de se ler porque o autor e o copista usaram as formas de numerais com que eles eram mais familiarizados. Livros produzidos em regiões próximas, ou separados por poucos anos, usaram caracteres diferentes para pelo menos alguns dos numerais. Foi a invenção da imprensa que os padronizou na forma em que nós os conhecemos agora, embora até hoje as formas do 5 e do 7 variem ligeiramente entre europeus e americanos. É interessante notar que, apesar do fato de que os europeus obtiveram o sistema dos árabes, as duas culturas utilizam formas de numerais notavelmente diferentes hoje. O turista europeu tem constantemente problemas com o fato de que o círculo é usado nos países árabes como símbolo para o dígito cinco e algo parecido com um ponto é usado para o símbolo do zero.

4.1.2 Desenvolvimentos iniciais da ciência do cálculo

Pode-se dizer que os primeiros passos em direção aos computadores digitais foram dados nas antigas civilizações da China, do Egito e da Babilônia, há mais de quatro milênios, com os sistemas de medidas de distâncias, previsão do curso das estrelas e tabelas gravadas em tábuas de barro usadas para ajudar cálculos algébricos. Durante a civilização grega algumas destas pré-ciências tomaram forma através dos *sistemas axiomáticos*[*]. Enquanto isso, é geralmente aceito que a Álgebra desenvolveu-se em cada civilização passando por sucessivas etapas, denominadas *retórica, sincopada e simbólica*[†].

Um museu em Oxford possui um cetro egípcio de mais de 5.000 anos, sobre o qual aparecem registros de 120.000 prisioneiros e 1.422.000 cabras capturadas **[Boy74]**. Apesar do exagero dos números, fica claro que os egípcios procuravam ser precisos no contar e no medir, bastando lembrar o alto grau de precisão das pirâmides. Medir as terras para fixar os limites das propriedades era uma tarefa importante nas civilizações antigas, especialmente no Egito. Ali, as enchentes anuais do Nilo, inundando as áreas férteis, derrubavam os marcos fixados no ano anterior, obrigando os proprietários de terras a refazer os limites de suas área de cultivo. Em algumas ocasiões, a questão era refazer os limites com base em informações parciais; conhecida a forma do terreno, tratava-se por exemplo de reconstruir os lados

---

[*] Em um sistema axiomático parte-se de premissas aceitas como verdadeiras e de regras ditas válidas, que irão conduzir a sentenças verdadeiras. As conclusões podem ser alcançadas manipulando-se símbolos de acordo com conjuntos de regras. A Geometria de Euclides é um clássico exemplo de um procedimento tornado possível por um sistema axiomático.

[†] "A álgebra retórica é caracterizada pela completa ausência de qualquer símbolo, exceto, naturalmente, que as próprias palavras estão sendo usadas no seu sentido simbólico. Nos dias de hoje esta álgebra retórica é usada em sentenças do tipo 'a soma é independente da ordem dos termos', que em símbolos seria designada por 'a+b = b+a' "**[Dan54]**. A sincopada é a notação intermediária que antecedeu a simbólica, caracterizada pelo uso de abreviações que foram sendo contraídas até se tornarem um símbolo **[idem]**.

restantes se um deles se havia preservado. Em outras ocasiões, destruídas por completo as fronteiras, tratava-se de refazê-las, de modo a remarcar o desejado número de propriedades, conservando as áreas relativas que possuíam no passado. Os egípcios tornaram-se hábeis delimitadores de terra e devem ter descoberto e utilizado inúmeros princípios úteis, relativos às características de linhas, ângulos e figuras – como por exemplo, o de que a soma de três ângulos de um triângulo é igual a dois ângulos retos, e o de que a área de um paralelogramo é igual à do retângulo que possua a mesma base e a mesma altura.

Provavelmente os egípcios obtiveram esses princípios por intermédio de raciocínios indutivos, fruto da observação e experimentação: mediam muitos triângulos e ângulos retos, áreas de muitos paralelogramos e retângulos, e parece que tais conhecimentos limitaram-se a habilitar os egípcios a resolver problemas de traçados de limites, de comparação de áreas, de projetos arquitetônicos e de engenharia de construções. No Egito antigo e na Babilônia existiam calculadores profissionais chamados *escribas* pelos egípcios e *logísticos* pelos gregos. As primeiras tentativas de invenção de dispositivos mecânicos para ajudar a fazer cálculos datam dessas épocas, como, por exemplo, o ábaco e o mecanismo Antikythera, sobre os quais se falará mais detidamente no capítulo da Pré-História Tecnológica.

Os gregos assimilaram os princípios empíricos dos egípcios e deram, a esse delimitado conhecimento, o nome de Geometria, isto é, medida da terra. Mas diferentemente daqueles, estudaram a Geometria mais sob seu aspecto teórico, desejando compreender o assunto por ele mesmo, independentemente de sua utilidade. Procuraram encontrar demonstrações dedutivas rigorosas das leis acerca do espaço e mostraram um crescente interesse pelos princípios geométricos. Pitágoras considerava que, em sua forma pura, a geometria se aproximava bastante da religião e para ele era o *arché*, o princípio de tudo, buscado tão intensamente pelos filósofos cosmológicos**[Bar67]**. Com a obra *Elementos*, de Euclides, reúnem-se e são apresentados de modo sistemático as principais descobertas geométricas de seus precursores, sendo considerado, até o século XIX, não somente o livro-texto da Geometria, mas o modelo daquilo que o pensamento científico deveria ser.

Resumindo, deve-se ver nestes tempos as tentativas de conceituação do número, o estabelecimento das bases numéricas, o estudo da Álgebra e da geometria e a busca de uma sistematização do raciocínio, que tanto atraíram os antigos. Tempos de evolução lenta e, em termos de produção efetiva de conhecimento matemático, bem abaixo da quantidade e qualidade produzida quase que exponencialmente a partir do século XV d.C., mas não menos importantes. De fato, para se compreender a História da Matemática na Europa é necessário conhecer sua história na Mesopotâmia e no Egito, na Grécia antiga e na civilização islâmica dos séculos IX a XV.

4.1.3 A Lógica de Aristóteles

Aristóteles (384 a.C. - 322 a.C.) é um filósofo 'atual': a 'questão aristotélica', isto é, o que Aristóteles realmente escreveu , o que se deve a ele ou antes, a seus discípulos, é algo complexo, onde não há acordo definitivo, e provavelmente segundo alguns nunca haverá, sendo uma questão sobre a qual se continua escrevendo. Aristóteles passou quase vinte anos na Academia platônica e educou Alexandre Magno (se Platão estivesse vivo teria visto a

realização de seu maior sonho, o de que os governantes filosofassem). Preocupava-o, como a todos os gregos, a vida política, a cidade, porém o que mais o interessava era o saber. E foi sábio em quase todos os domínios: ciências naturais, lógica, física, poética, astronomia, ética, política, retórica, psicologia, entre outras. Mestre de lógica para centenas de gerações, aplicou-se, sobretudo, em assentar as bases da "ciência que buscamos", a "filosofia primeira", o que depois chamou-se Metafísica. Interessa neste estudo sobretudo a lógica aristotélica e o seu método axiomático.

A Lógica foi considerada na tradição clássica e medieval como instrumento indispensável ao pensamento científico. Atualmente é parte importante na metodologia dedutiva das ciências, além de constituir-se como um saber próprio, com ligação a relevantes problemas teóricos. Da Lógica Científica nasceu a Lógica Matemática e, dentro desta, várias filosofias da lógica que interpretam os cálculos simbólicos e sua sistematização axiomática. Para a História da Computação interessa abordar em particular a questão do pensamento dedutivo e matemático, seus limites, o problema da relativa mecanização do pensamento quantitativo e o problema da Inteligência Artificial. Da discussão e busca da solução desses problemas, que entram também no campo filosófico, formou-se a base conceitual, *teoria da computabilidade*, necessária para o advento do computadores.

O início da ciência da Lógica encontra-se na antiga Grécia **[Kne68] [Boc66]**. As polêmicas geradas pela teoria de Parmênides e os famosos argumentos de Zenão*, que negavam a realidade do movimento fazendo um uso indevido do princípio da não-contradição, contribuíram para a distinção dos conceitos, para se ver a necessidade de argumentar com clareza, mediante demonstrações rigorosas, e assim responder às objeções dos adversários. Mais tarde, as sutilezas dos *sofistas*, que reduziam todo o saber à arte de convencer pelas palavras, levaram Sócrates a defender o valor dos conceitos e tentar defini-los com precisão. Assim a Lógica como ciência vai se formando pouco a pouco, principalmente com Sócrates e Platão. Mas Platão pensava que qualquer conteúdo da mente existia tal qual na realidade e Aristóteles reage ao seu mestre, dizendo que as idéias existem somente na mente humana, mas correspondendo a realidades.

Com Aristóteles é que se dá o verdadeiro nascimento da Lógica como ciência das idéias e dos processos da mente. "Até hoje não existe forma alguma concebível de lógica, por muito distinta que seja da lógica formal, que não tenha algum tipo de conexão com a obra aristotélica" **[Sch31]**. Ele foi o primeiro lógico formal da história, tendo desenvolvido ao menos duas formas distintas de lógica formal, elaborando algumas de suas partes de maneira praticamente completa e deixando esboçados outros tipos de lógicas que somente na época atual foram novamente tratadas†.

---

* Parmênides (540 a 470 a.C.) negava a existência do movimento ("devir") e afirmava a existência de um único ser (panteísmo), tendo enunciado o princípio da não contradição: 'algo não pode ser e não ser ao mesmo tempo, sob o mesmo aspecto e no mesmo sujeito'. Seu discípulo Zenão (490 a 430 a.C.) foi o fundador da dialética e radicalizou a negação do movimento. Este envolveria um paradoxo: para mudar completamente é preciso antes mudar parcialmente, e assim infinitamente, o que levaria a concluir que o movimento não existe (paradoxos de Aquiles e a tartaruga e os pontos de percurso de uma flecha)

† Além do mais interessa o estudo das obras do Filósofo pois tem um especial valor pedagógico, ao apresentar de maneira unitária a maior parte dos problemas lógicos, contemplados com o vigor característico que acompanha uma ciência emergente, e mais acessível ao principiante que muitas apresentações modernas de lógica formal.

Aristóteles escreveu uma série de trabalhos que seriam editados por Andrônico de Rodes no século I d.C. e que receberam posteriormente o nome de *Organon* ("Instrumento"), de acordo com a concepção segundo a qual a Lógica deveria fornecer os instrumentos mentais necessários para enfrentar qualquer tipo de investigação. Essa obra compreende os seguintes livros: *Categorias, Analíticos I, Analíticos II,* o *Peri Hermeneias* (ou sobre a interpretação), *Tópicos* e *Refutação de argumentos sofistas.* A grande novidade aristotélica está nos Analíticos, com o silogismo. Aristóteles chamava a Lógica com o termo "analítica" (e justamente "Analíticos" são intitulados os escritos fundamentais do Organon). A analítica (do grego *analysis*, que significa "resolução") explica o método pelo qual, partindo de uma dada conclusão, resolve-se precisamente nos elementos dos quais deriva, isto é, nas premissas e nos elementos de que brota, e assim fica fundamentada e justificada.

Aristóteles construiu uma sofisticada teoria dos argumentos, cujo núcleo é a caracterização e análise dos chamados *silogismos*, os típicos raciocínios da lógica aristotélica. O argumento

Todo homem é mortal
Sócrates é homem
Logo Sócrates é mortal

é o exemplo típico do silogismo perfeito. Conforme o próprio Aristóteles, "o silogismo é um discurso no qual, sendo admitidas algumas coisas, outra coisa distinta resulta necessariamente dessas coisas afirmadas primeiro, pelo único fato de que essas existem" **[Per88]**.

Figura 4: Aristóteles

Nos Primeiros Analíticos, Aristóteles desenvolveu minuciosamente o sistema dos silogismos, mostrando os princípios maiores que o sustentam e as regras que lhe devem

moldar a construção. A análise do Filósofo é tão ampla quanto engenhosa e envolve também as assim chamadas "modalidades" e os silogismos modais*.

Entre as características mais importantes da silogística aristotélica está a de se ter pensado pela primeira vez na história da lógica em fazer uso de letras que poderiam ser usadas para representar uma expressão substantiva qualquer, fundamental para o desenvolvimento do simbolismo lógico. É também com Aristóteles que se encontra uma das primeiras tentativas de se estabelecer um rigor nas demonstrações matemáticas. Ao definir os dois tipos de demonstração, *quia* (dos efeitos às causas) e *propter quid* (das causas aos efeitos), dizia (I Anal. Post., lect. 14) que as matemáticas utilizam preferencialmente esse modo de demonstrar, e por isso esta ciência é essencialmente dedutiva: "algumas vezes o mais conhecido por nós em si mesmo e por natureza é também o mais cognoscível em si mesmo e por natureza. Assim acontece nas matemáticas, nas quais, devido à abstração da matéria, não se efetuam demonstrações mais do que a partir dos princípios formais. E assim as demonstrações procedem desde o mais cognoscível em si mesmo".

No entanto, para que haja demonstração, os primeiros princípios devem ser indemonstráveis, já que, do contrário, se procederia a uma regressão ao infinito. E como se conhece a verdade dos primeiros princípios? Por indução. "À força de contemplar a freqüência dos acontecimentos, buscando o universal". Os primeiros princípios não são inatos; são adquiridos na experiência, analisando-se as percepções que contém um elemento universal. No entanto esta 'indução' citada por Aristóteles nada tem a ver com 'demonstração por indução completa' ou o 'método da indução finita' ou, mais ainda, 'raciocínio por recorrência', estabelecido pelo matemático italiano Giuseppe Peano. Sobre esta temática ver anexo sobre *Dedução e Indução na Matemática*.

E no meio de tudo isso emergiu aquela que é considerada uma das mais definitivas contribuições do Organon: o método axiomático, que recebeu precisamente no século XX sua maior valoração, ao apresentar-se como o caminho universal dentro do qual se enquadram todas as ciências dedutivas. Um pouco mais sobre este assunto é exposto no anexo *O método axiomático e as ciências dedutivas*.

4.1.4 A contribuição dos megáricos e estóicos

Embora Aristóteles seja o mais brilhante e influente filósofo grego, outra importante tradição argumentativa formou-se na antiga Grécia, com os megáricos e estóicos. Pouco conservada pela tradição, merece um melhor tratamento dos historiadores, porque o que deles se conhece sugere que esses gregos eram altamente inteligentes.

Os megáricos (em função de sua cidade, Mégara) interessaram-se por certos enigmas lógicos como o conhecido "paradoxo do mentiroso": quem diz "O que eu afirmo agora é falso", enuncia algo verdadeiro ou falso? Um deles, Diodoro Cronus, que morreu por volta de 307 a.C., formulou interessante concepção modal, relacionando possibilidade, tempo e verdade, enquanto outro megárico, de nome Fílon, estudou proposições do tipo "Se chove então a rua está molhada", construída com o auxílio das expressões "se..., então..." conhecidas

* Modalidades são as expressões do tipo "é possível que...", "é necessário que...".

como condicionais. Ele as definiu em termos extremamente polêmicos, mas que seriam assumidos como corretos, vinte e três séculos mais tarde pelos fundadores da Lógica Contemporânea.

Os estóicos (da chamada escola filosófica de "Stoa", que quer dizer "pórtico") desenvolveram também notáveis teorias lógicas. Tinham bastante presente a diferença que há entre um código de comunicação específico, de um lado, e o que se pode expressar através do uso de tal código. Assim sendo, um conceito de "proposição" análogo ao usado na atual Lógica, já estava presente, de modo virtual, na filosofia estóica da linguagem.

Porém a mais notável contribuição estóica à Lógica foi obra de Crísipo de Soles (280-206 a.C.), homem de vasta produção poligráfica (750 livros). Ele estudou as sentenças condicionais e também as disjuntivas (regidas pela partícula "ou") e as copulativas (regidas pelo "e"), tendo também reconhecido claramente o papel lógico desempenhado pela negação. Além disto, Crísipo foi capaz de relacionar tais idéias com as modalidades, elaborando, então, um sistema de princípios lógicos que, no seu campo específico, foi muito além dos poucos resultados obtidos por Aristóteles e seu discípulo Teofrasto. Por tal razão, Crísipo é reconhecido como o grande precursor daquilo que hoje se chama "Cálculo Proposicional", o primeiro capítulo da Lógica desenvolvida a partir do último quarto do século XIX **[Bri79b]**.

4.1.5 Euclides e o Método Axiomático

Figura 5: Representação de Euclides

Com sua obra *Elementos*, o matemático grego Euclides (330 a.C. - 277 a.C.) deu uma forma sistemática ao saber geométrico, implementando as idéias sobre axiomatização, de Aristóteles, para uma ciência exata. No primeiro livro dos Elementos, ele enuncia vinte e três definições, cinco postulados e algumas noções comuns ou axiomas*. Em seguida ele deduz proposições ou teoremas, os quais constituem o saber geométrico, como por exemplo: "se em um triângulo dois ângulos são iguais entre si, também os lados opostos a esses ângulos são iguais entre si". Postulados, axiomas e definições constituem os pontos de partida para as demonstrações de Euclides. Seu objetivo é mostrar todos os outros princípios geométricos – primeiro os da Geometria Plana e depois os da Geometria Espacial –, revelando que são decorrências necessárias dos princípios fundamentais.

Quais são os traços característicos das técnicas adotadas por Euclides? Em primeiro lugar ele enuncia as sua leis em forma universal: não se detém em determinada figura ou linha, mas examina as propriedade que todas as figuras e todas as linhas de tal ou qual tipo devem ter. Formula tais leis de maneira rigorosa e absoluta e, mais ainda, demonstra-as. Seu livro, na verdade, consiste em demonstrações colocadas de maneira sistemática, não indutiva, mas dedutiva, por meio das quais procura estabelecer as suas conclusões com o rigor da lógica. Euclides visava aperfeiçoar o conhecimento acerca de pontos, linhas e figuras, tornando mais rigorosas as demonstrações de leis já conhecidas, e procurava aumentar esse conhecimento, demonstrando leis novas, até então desconhecidas. Mas talvez não se esgotasse aí a motivação – ou pelo menos as conseqüências – do que elaborou o geômetra. A colocação de axiomas e teoremas em forma dedutiva deu à Geometria uma apresentação mais elegante e transparente, tornando facilmente perceptíveis as interessantes conexões lógicas ali introduzidas. A axiomatização do saber Geométrico abriu um sem fim de perspectivas para os estudiosos das ciências exatas, que adotam as exposições axiomáticas – e buscam axiomatizações mais elegantes e econômicas – não só para dar rigor às suas demonstrações, mas descobrir novas conexões lógicas.

Esse é portanto o modo como Euclides ordena o conhecimento geométrico no chamado sistema euclidiano. Durante séculos esse sistema valeu como modelo insuperável do saber dedutivo: os termos da teoria são introduzidos depois de terem sido definidos e as proposições não são aceitas se não forem demonstradas. Euclides escolhia as *proposições primitivas*, base da cadeia sobre a qual se desenvolvem as deduções sucessivas, de tal modo que ninguém pudesse levantar dúvidas sobre a sua veracidade: eram auto-evidentes, portanto isentas de demonstração. Leibniz afirmaria mais tarde que os gregos raciocinavam com toda a exatidão possível em matemática e deixaram à humanidade modelos de arte demonstrativa **([RA91], volume III)**.

Em resumo, Euclides, como já fizera Aristóteles, buscou o ideal de uma organização axiomática, que em última instância se reduz à escolha de um pequeno número de proposições em princípio aceitas naquele domínio do conhecimento, e à posterior dedução de todas as outras proposições verdadeiras desse domínio, a partir delas. Surge com Euclides e

* As definições pretendem substancialmente explicitar os conceitos da geometria ("ponto é aquilo que não tem partes"; "linha é comprimento sem largura", etc.). Os postulados representam verdades indubitáveis típicas do saber geométrico ("pode-se levar uma reta de qualquer ponto a qualquer ponto"; "todos os ângulos retos são iguais"; etc.). Os axiomas são verdades que valem universalmente, não só na geometria ("o todo é maior que a parte"; "coisas que são iguais a uma mesma coisa são iguais entre si", etc.).

Aristóteles (estará plenamente desenvolvida no início do século XX com a escola formalista de Hilbert) a busca de uma *economia do pensamento* (um bom texto sobre o assunto pode ser encontrado em **[Wil65]**). A História da Computação tem um marco significativo nesse ponto da História: o começo da busca da automatização do raciocínio e do cálculo.

Mas havia um problema no sistema de Euclides: suas "evidências" não eram assim tão evidentes. O seu quinto postulado não convenceu de modo algum, e despertou perplexidade na história do próprio pensamento grego, depois no árabe e no renascentista. No século XIX, Karl Friedrich Gauss (1777-1855) viu com toda a clareza a não demonstrabilidade do quinto postulado e a possibilidade da construção de sistemas geométricos não euclidianos. Janos Boulay (1802-1860), húngaro, e Nicolai Ivanovic Lobacewskiy (1793-1856), russo, trabalhando independentemente, elaboraram uma geometria na qual o postulado da paralela não vale mais.

A conseqüência desses fatos foi a eliminação dos poderes da intuição na fundamentação e elaboração de uma teoria geométrica: os axiomas não são mais "verdades evidentes" que garantem a "fundação" do sistema geométrico, mas puros e simples pontos de partida, escolhidos convencionalmente para realizar uma construção dedutiva. Mas, se os axiomas são puros pontos de partida, quem garantirá que, continuando-se a deduzir teoremas, não se cairá em contradição?

Esta questão crucial dos fundamentos da matemática levará aos grandes estudos dos finais do século XIX e inícios do XX e será o ponto de partida do projeto formalista de David Hilbert, assim como de outras tentativas de se fundamentar a matemática na lógica e na teoria dos conjuntos, como as propostas por Frege, Russell e Cantor. E será dessa seqüência de sucessos e fracassos que se produzirá a base da Computação, com Turing, von Neumann, Post, Church, e outros mais.

4.1.6 Diophantus, al-Kharazmi e o desenvolvimento da Álgebra

O seguinte problema no *Rhind Papyrus*[*], do Museu britânico em Londres, foi escrito por volta do ano 1650 a.C.:

> *Divida 100 pães entre 10 homens, incluindo um barqueiro, um capataz e um vigia, os quais recebem uma dupla porção cada. Quanto cabe a cada um?* ***[Bow94]***

Isto naturalmente pode ser resolvido usando-se Álgebra.

O primeiro tratado de Álgebra foi escrito pelo grego Diophantus (200 - 284), da cidade de Alexandria, por volta do ano 250. O seu *Arithmetica,* composto originalmente por 13 livros dos quais somente 6 se preservaram, era um tratado "caracterizado por um alto grau de habilidade matemática e de engenho: quanto a isto, o livro pode ser comparado aos grandes clássicos da idade alexandrina anterior" **[Boy74]**. Antes de Diophantus, toda a

[*] Juntamente com o papiro de Moscou é uma das principais fontes de informação relativa às formas de notação e operações aritméticas em uso durante a primeira época da civilização egípcia

'álgebra' que havia, incluindo problemas, operações, lógica e solução, era expressada sem *simbolismo* – palavra chave sobre a qual ainda se voltará a falar –; ele foi o primeiro a introduzir o simbolismo na matemática grega. Para uma quantidade desconhecida usava um símbolo (chamado *arithmos*), que caracterizava um número indefinido de unidades. Pela ênfase dada em seu tratado à solução de problemas indeterminados, tal tratado tornou-se conhecido como *análise diofantina*, que em geral faz parte da disciplina de Teoria dos Números*. Seu trabalho, contudo, não é suficiente para lhe conferir o título de pai da Álgebra†.

Mas é com os persas e principalmente com os árabes que a Álgebra poderá ser efetivamente chamada de ciência. É interessante notar que ao se falar que a Geometria é uma ciência grega ou que a Álgebra é uma ciência árabe, está se afirmando algo mais do que a "casualidade" de terem sido gregos ou árabes seus fundadores ou promotores. Ordinariamente tendemos a pensar que o conhecimento científico independe de latitudes e culturas: uma fórmula química ou um teorema de Geometria são os mesmos em inglês ou português ou chinês e, sendo a comunicação, à primeira vista, o único problema, bastaria uma boa tradução dos termos próprios de cada disciplina. Mas não é assim.

Na verdade a evolução da ciência está repleta de interferências histórico-culturais, condicionando metodologias, o surgimento de novas áreas do saber, e assim por diante. Os juristas árabes referem-se à Álgebra como "o cálculo da herança", segundo a lei corânica, uma problemática importante dentro do Islam, e aí já temos um exemplo de condicionamento histórico-cultural. Não foi por mero acaso que a Álgebra surgiu no califado abássida ("ao contrário dos Omíadas, os Abássidas pretendem aplicar rigorosamente a lei religiosa à vida cotidiana" **[AG81]**), no seio da "Casa da Sabedoria" (*Bayt al-Hikma*) de Bagdá, promovida pelo califa Al-Ma'amun; uma ciência nascida em língua árabe e antagônica da ciência grega. Embora hoje a Álgebra possa parecer objetiva e axiomática, com uma sintaxe de estruturas operatórias e destituída de qualquer alcance semântico, ela é o resultado da evolução da velha *al'jabr*, "forjada por um contexto cultural em que não são alheios elementos que vão desde as estruturas gramaticais do árabe à teologia muçulmana da época" **[Lau97]**.

Muhammad Ibn Musa Al-Khwarizmi (780 - 850), matemático e astrônomo persa, foi membro da "Casa da Sabedoria", a importante academia científica de Bagdá, que alcançou seu resplendor com Al-Ma'amun (califa de 813 a 833). A ele, Al-Khwarizmi dedicou seu *Al-Kitab al-muhtasar fy hisab al-jabr wa al-muqabalah* ("Livro breve para o cálculo da *jabr* e da *muqabalah*"). *Al-jabr,* que significa *força que obriga*, *restabelecer*‡, precisamente porque a Álgebra é "forçar cada termo a ocupar seu devido lugar". Já no começo do seu *Kitab*, Al-Khwarizmi distingue seis formas de equação, às quais toda equação pode ser *reduzida* (e, canonicamente resolvida). Na notação atual:

---

* Neste sentido foi o matemático grego que maior influência teve sobre a moderna teoria dos números. Em particular Fermat foi levado ao seu 'último' teorema quando procurou generalizar um problema que tinha lido na *Arithmetica* de Diophantus: dividir um dado quadrado em dois quadrados (ver F.E. Robbins, *P.Mich. 620: A Series of Arithmetical Problems,* Classical Philology, pg 321-329, EUA, 1929).

† Apenas sob certo aspecto isto se justificaria. Em uma visão um tanto arbitrária e simplista poderíamos dividir o desenvolvimento da álgebra em 3 estádios: (1) primitivo, onde tudo é escrito em palavras; (2) intermediário, em que são adotadas algumas simplificações; (3) simbólico ou final. Neste contexto, *Arithmetica* deve ser colocada na segunda categoria.

‡ Restabelecer, por algo no seu devido lugar, restabelecer uma normalidade. Como observa **[Lau97]** a palavra *tajbir* significa ortopedia e *jibarath* significa redução, no sentido médico (por o osso no devido lugar) e na Espanha, no tempo em que os barbeiros acumulavam funções, podia-se ver a placa "Algebrista e Sangrador" em barbearias.

1. $ax^2 = bx$
2. $ax^2 = c$
3. $ax = c$
4. $ax^2 + bx = c$
5. $ax^2 + x = ax^2$
6. $bx + c = ax^2$

Al-jabr é a operação que soma um mesmo fator (afetado do sinal +) a ambos os membros de uma equação, para eliminar um fator afetado pelo sinal -. Já a operação que elimina termos iguais de ambos os lados da equação é *al-muqabalah* que significa estar de frente, cara a cara, confrontar. Por exemplo: em um problema onde os dados podem ser colocados sob a forma $2x^2 + 100 – 20x = 58$, Al-Khwarizmi procede do seguinte modo:
$2x^2 + 100 = 58 + 20x$ (por *al-jabr*)
Divide por 2 e reduz os termos semelhantes:
$x^2 + 21 = 10\,x$ (por *al-muqabalah*)

E o problema já está canonicamente equacionado.

Figura 6: Representação de Al-Kharazmi

Al-Kharazmi introduziu a escrita dos cálculos no lugar do uso do ábaco. De seu nome derivaram as palavras, como já citado acima na história do desenvolvimento do conceito de número, *algarismo* e *algoritmo*[*] **[Kar61]**.

Embora não muito visível ainda, deve-se chamar a atenção para essa disciplina da Álgebra, que deve ser colocada entre as ciências que fundamentaram o desenvolvimento da

[*] A palavra algoritmo na matemática designa um procedimento geral de cálculo, que se desenvolve, por assim dizer, automaticamente, poupando-nos esforço mental durante o seu curso; este termo será depurado e aproveitado dentro da Computação. Dele se tornará a falar mais à frente

Computação. Pois o computador e todos os instrumentos que o precederam (réguas de cálculo, máquina de Pascal, a calculadora de Leibniz, a máquina analítica de Babbage, etc.) são somente as manifestações práticas que foram surgindo, com naturalidade, em resultado da busca pelo homem de reduzir os problemas a expressões matemáticas, resolvendo-as segundo regras. E isto, há muitos séculos, já tinha tomado o nome de Álgebra, a "arte dos raciocínios perfeitos" como dizia Bhaskara, o conhecido matemático hindu do século XII. Com os árabes, depois de relativo obscurecimento da cultura grega, dá-se continuidade ao processo que proporcionará as bases fundamentais para o raciocínio automatizado, fundamental na Ciência da Computação.

4.1.7 A automatização do raciocínio

Ainda dentro do período acima estabelecido (4.200 a.C. até meados do ano 1600 d.C) iniciou-se concretização de uma antiga meta: a idéia de se reduzir todo raciocínio a um processo mecânico, baseado em algum tipo de cálculo formal. Isto remonta a Raimundo Lúlio. Embora negligenciado pela ciência moderna, Raimundo Lúlio (1235 - 1316), espanhol, figura pletórica de seu tempo, em seu trabalho *Ars Magna* (1305 - 1308), apresentou a primeira tentativa de um procedimento mecânico para produzir sentenças logicamente corretas **[Her69]**. Lúlio acreditava que tinha encontrado um método que permitia, entre outras coisas, tirar todo tipo de conclusões, mediante um sistema de anéis circulares dispostos concentricamente, de diferentes tamanhos e graduáveis entre si, com letras em suas bordas. Invenção única, tentará cobrir e gerar, representando com letras – que seriam *categorias do conhecimento* –, todo o saber humano, sistematizado em uma gramática lógica.

Figura 7: Desenho de Raimundo Lúlio

Conforme Hegel: "A tendência fundamental da 'arte' de Raimundo Lúlio consistia em enumerar e ordenar todas as determinações conceituais a que era possível reduzir todos os objetos, as categorias puras com referência às quais podiam ser determinados, para, desse

modo, poder assinalar facilmente, com respeito a cada objeto, os conceitos a ele aplicáveis. Raimundo Lúlio é, pois, um pensador sistemático, ainda que ao mesmo tempo mecânico. Deixou traçada uma tabela em círculos nos quais se acham inscritos triângulos cortados por outros círculos. Dentro desses círculos, ordenava as determinações conceituais, com pretensões exaustivas; uma parte dos círculos é imóvel, a outra tem movimento. Vemos, com efeito, seis círculos, dois dos quais indicam os sujeitos, três os predicados e o sexto as possíveis perguntas. Dedica nove determinações a classe, designando-as com as nove letras B C D E F G H I K. Obtém, desse modo, nove predicados absolutos, que aparecem escritos ao redor de seu quadro: a bondade, a magnitude, a duração, o poder, a sabedoria, a vontade, a virtude, a verdade, a magnificência; em seguida, vêm nove predicados relativos: a diferença, a unanimidade, a contraposição, o princípio, a metade, o fim, o ser maior, o ser igual e o ser menor; em terceiro lugar, temos as perguntas: sim?, quê?, de onde?, por quê?, quão grande?, de que qualidade?, quando?, de onde?, como e com quê?, a última das quais encerra duas determinações; em quarto lugar, aparecem nove substâncias (*esse*), a saber: Deus (*divinum*), os anjos (*angelicum*), o céu (*coeleste*), o homem (*humanum*), *Imaginativum*, *Sensitivum*, *Vegetativum*, *Elementativum*; em quinto lugar, nove acidentes, quer dizer, nove critérios naturais: a quantidade, a qualidade, a relação, a atividade, a paixão, o ter, a situação, o tempo e o lugar; por último, nove critérios morais, que são as virtudes: a justiça, a prudência, a valentia, a temperança, a fé, a esperança, o amor, a paciência e a piedade, e nove vícios: a inveja, a cólera, a inconstância, a avareza, a mentira, a gula, a devassidão, o orgulho e a preguiça (*acedia*). Todos esses círculos tinham de ser colocados necessariamente de determinado modo para poder dar como resultado as combinações desejadas. Conforme as regras de colocação, segundo as quais todas as substâncias recebem os predicados absolutos e relativos adequados a estes, deviam ser esgotados a ciência geral, a verdade e o conhecimento de todos os objetos concretos." **[Roc81]**.

Os procedimentos estabelecidos por Lúlio não foram muito válidos. Mas o mais importante em Lúlio é a idéia concebida, genial sob certo aspecto. Tanto que seu trabalho influenciará muitos matemáticos famosos, do nível de um Cardano (século XVI), Descartes (século XVII), Leibniz (séculos XVII/XVIII), Cantor (séculos XIX/XX), entre outros. Raimundo Lúlio é considerado o precursor da análise combinatória. Como dirá R. Blanché: "encontramos em Lúlio, pelos menos em germe e por mais que ele não soubesse tirar partido disso por inabilidade, duas idéias que iriam se tornar predominantes nas obras de Lógica, primeiro em Leibniz e depois em nossos contemporâneos: as idéias de característica e de cálculo (...). Com a ajuda desse simbolismo, tais autores pretendem permitir que as operações mentais, freqüentemente incertas, fossem substituídas pela segurança de operações quase mecânicas, propostas de uma vez por todas" **([RA91], volume III).**

Pode-se ver em Raimundo Lúlio os primórdios do desenvolvimento da Lógica Matemática, isto é, de um novo tratamento da ciência da Lógica: o procurar dar-lhe uma forma matemática. O interesse deste trabalho é caracterizar a Lógica Matemática, sem aprofundar nas discussões filosóficas – ainda em aberto – sobre os conceitos "lógica matemática" e "lógica simbólica", se é uma lógica distinta da ciência matemática ou não, pois sem dúvida alguma a Computação emergirá dentro de um contexto da evolução deste novo tratamento da lógica.

A Lógica Matemática ergue-se a partir de duas idéias metodológicas essencialmente diferentes. Por um lado é um cálculo, daí sua conexão com a matemática. Por outro lado, caracteriza-se também pela idéia de uma *demonstração exata* e, nesse sentido, não é uma imitação da matemática nem esta lhe serve de modelo, mas pelo contrário, à Lógica caberá investigar os fundamentos da matemática com métodos precisos e oferecer-lhe o instrumento para uma demonstração rigorosa.

A palavra Álgebra voltará a aparecer com o inglês Robert Recorde(1510?-1558), em sua obra *Pathway of Knowledge*(1551), que introduz o sinal de '=' e divulga os símbolos '+' e '–', introduzidos por John Widmann (Arithmetica, Leipzig, 1489). Thomas Harriot(1560-1621) prosseguirá o trabalho de Recorde, inventando os sinais '>' e '<'. Willian Oughtred(1574-1621), inventor da régua de cálculo baseada nos logaritmos de Napier, divulgou o uso do sinal '×', tendo introduzido os termos seno, coseno e tangente. Em 1659 J.H. Rahn usou o sinal '÷'. Todos esses matemáticos ajudaram a dar à Álgebra sua forma mais moderna.

BC	CD	DE	EF	FG	GH	HI	IK
BD	CE	DF	EG	FH	GI	HK	
BE	CF	DG	EH	FI	GK		
BF	CG	DH	EI	FK			
BG	CH	DI	EK				
BH	CI	DK					
BI	CK						
BK							

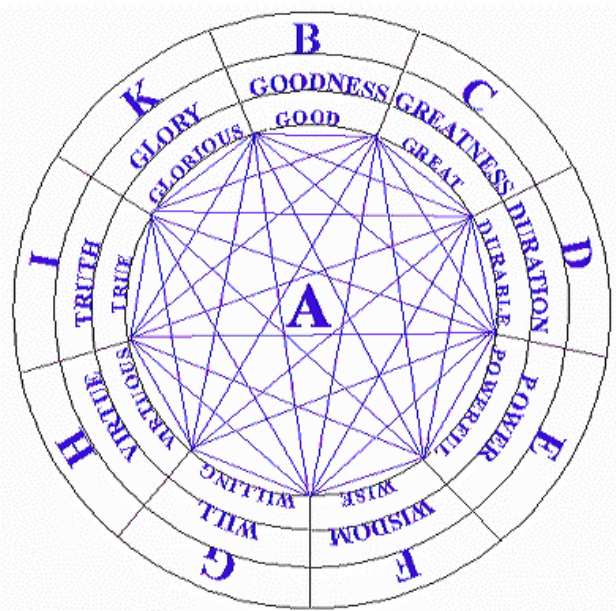
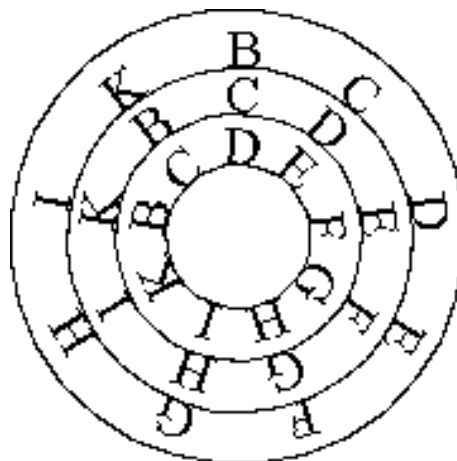
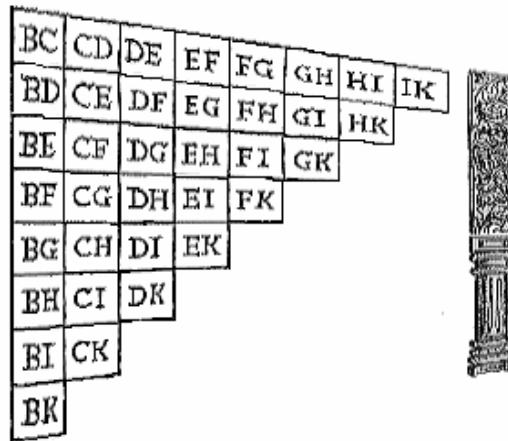
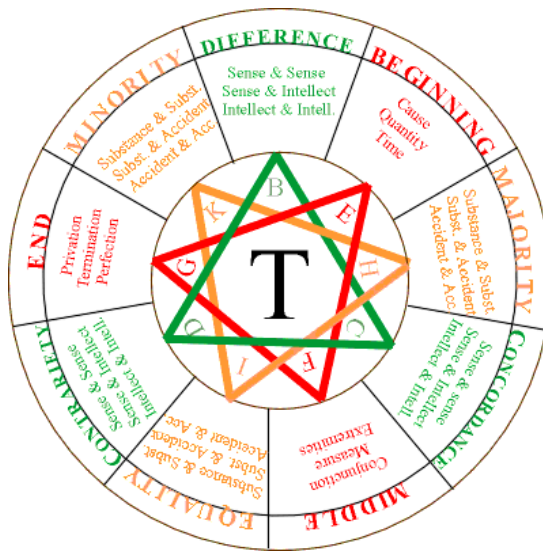



Figura 8: Figuras representando mecanismo elaborado por Lúlio para automatizar o raciocínio

4.2 A mecanização do cálculo

Se é aceito o ponto de vista de estudiosos como Needham **[Nee59]**, a data de 1600 pode ser vista como um bom divisor de águas dentro da história da ciência em geral. Vale a pena lembrar que o estudo da matemática no tempo anterior a essa data, na Europa, não havia avançado substancialmente em relação ao mundo árabe, hindu ou chinês.

A álgebra árabe fora perfeitamente dominada e tinha sido aperfeiçoada, e a trigonometria se tornara uma disciplina independente **[Boy74]**. O casamento de ambas pela aplicação dos métodos algébricos no terreno da geometria foi o grande passo e Galileu (1564 - 1642) aí tem um papel preponderante. Ele uniu o experimental ao matemático, dando início à ciência moderna. Galileu dá uma contribuição decisiva a uma formulação matemática das ciências físicas. A partir de então, em resultado desse encontro da matemática com a física, a ciência tomou um novo rumo, a um passo mais rápido, e rapidamente as descobertas de Newton (1643 -1727) sucedem às de Galileu.

Trata-se de um período de transição por excelência, que preparou o caminho para uma nova matemática: não já uma coleção de truques, como Diophantus possuíra, mas uma forma de raciocinar, com uma notação clara. É o começo do desenvolvimento da idéia de formalismo na Matemática, tão importante depois para a fundamentação teórica da Computação*.

4.2.1 Leibniz, o precursor da Lógica Matemática moderna

A Lógica Moderna começou no século XVII com o filósofo e matemático alemão Gottfried Wilhelm Leibniz (1646 - 1716). Seus estudos influenciaram, 200 anos mais tarde, vários ramos da Lógica Matemática moderna e outras áreas relacionadas, como por exemplo a Cibernética (Norbert Wiener dizia que "se fosse escolher na História da Ciência um patrono para a Cibernética, elegeria Leibniz" **[Wie70]**).

Entre outras coisas, Leibniz queria dotar a Metafísica (aquela parte da Filosofia que estuda o "ser" em si) de um instrumento suficientemente poderoso que a permitisse alcançar o mesmo grau de rigor que tinha alcançado a Matemática. Parecia-lhe que o problema das interrogações e polêmicas não resolvidas nas discussões filosóficas, assim como a insegurança dos resultados, eram fundamentalmente imputáveis à ambigüidade dos termos e dos processos conclusivos da linguagem ordinária. Leibniz tentaria elaborar sua nova lógica precisamente como projeto de criação de uma lógica simbólica e de caráter completamente calculístico, análogos aos procedimentos matemáticos.

Historicamente falando, tal idéia já vinha sendo amadurecida, depois dos rápidos desenvolvimentos da Matemática nos séculos XVI e XVII, possibilitados pela introdução do simbolismo. Os algebristas italianos do século XVI já tinham encontrado a fórmula geral

* Lembrando algo que já foi dito, é importante ressaltar que desde suas origens aristotélicas a lógica havia assumido claramente alguns recursos fundamentais, como a estrutura formal, o emprego de certo grau de simbolismo, a sistematização axiomática e o identificar-se com a tarefa de determinar as "leis" do discurso (tomando, por exemplo, a linguagem como tema de estudo), características que foram assumidas pela Lógica Moderna.

para a resolução das equações de terceiro e quarto graus, oferecendo à Matemática um método geral que tinha sido exaustivamente buscado pelos antigos e pelos árabes medievais. Descartes e Fermat criaram a geometria analítica, e, depois de iniciado por Galileu, o cálculo infinitesimal desenvolveu-se com grande rapidez, graças a Newton e ao próprio Leibniz. Ou seja, as matemáticas romperam uma tradição multissecular que as havia encerrado no âmbito da geometria, e estava se construindo um simbolismo cada vez mais fácil de manejar e seguro, capaz de funcionar de uma maneira, por assim dizer, mecânica e automática, sujeito a operações que, no fundo, não eram mais do que regras para manipulação de símbolos, sem necessidade de fazer uma contínua referência a conteúdos geométricos intuitivos.

Figura 9: Leibniz

Leibniz deu-se conta de tudo isto e concebeu, também para a dedução lógica, uma desvinculação análoga com respeito ao conteúdo semântico das proposições, a qual além de aliviar o processo de inferência do esforço de manter presente o significado e as condições de verdade da argumentação, pusesse a dedução a salvo da fácil influência que sobre ela pode exercer o aspecto material das proposições. Deste modo coube a ele a descoberta da verdadeira natureza do "cálculo" em geral, além de aproveitar pela primeira vez a oportunidade de reduzir as regras da dedução lógica a meras regras de cálculo, isto é, a regras cuja aplicação possa prescindir da consideração do conteúdo semântico das expressões.

Leibniz influenciou seus contemporâneos e sucessores através de seu ambicioso programa para a Lógica. Esse programa visava criar uma linguagem universal baseada em um alfabeto do pensamento ou *characteristica universalis*, uma espécie de cálculo universal para o raciocínio.

Figura 10: Blaise Pascal

Na visão de Leibniz a linguagem universal deveria ser como a Álgebra ou como uma versão dos ideogramas chineses: uma coleção de sinais básicos que padronizassem noções simples não analíticas. Noções mais complexas teriam seu significado através de construções apropriadas envolvendo sinais básicos, que iriam assim refletir a estrutura das noções complexas e, na análise final, a realidade. O uso de numerais para representar noções não analíticas poderia tornar possível que as verdades de qualquer ciência pudessem ser "calculadas" por operações aritméticas, desde que formuladas na referida linguagem universal **([Bri79a], volume XI)**. Com isso se poderia substituir o genérico dialoguemos por um mais exato calculemos. Conforme o próprio Leibniz, "Quando orietur controversiae, non magis disputatione opus erit inter duo philosophos, quam inter duo computistas. Sufficet enin calamos in manus sumere, sedereque ad ábacos et sib mutuo (accito si placet amico) dicere: calculemus"*. As discussões não seriam, assim, disputas controvertidas, de resultado duvidoso e final não concluído, mas sim formas de cálculo que estabelecessem a maior ou menor verdade de uma proposição.

Essa idéia de Leibniz sustentava-se em dois conceitos intimamente relacionados: o de um simbolismo universal e o de um cálculo de raciocínio (isto é, um método quase mecânico de raciocínio)†. Isso, para a História da Computação, tem um particular interesse, pois esse *calculus ratiocinator* de Leibniz contém o embrião da *machina ratiocinatrix*, a máquina de raciocinar buscada por Turing e depois pelos pesquisadores dentro do campo da Inteligência Artificial. Leibniz, assim como Boole, Turing, e outros – basta lembrar o ábaco, o

* Quando aparecer uma controvérsia, já não haverá necessidade de uma disputa entre dois filósofos mais do que a que há entre dois calculistas. Bastará, com efeito, tomar a pena na mão, sentar-se à mesa (ad abacus) e (ao convite de um amigo, caso se deseje), dizer um ao outro: Calculemos!" **[Boc66]**

† Deve-se observar que destas conceituações descenderam a notação da matemática e da lógica do século XX.

'computador' de Babbage, etc. –, perceberam a possibilidade da mecanização do cálculo aritmético. O próprio Leibniz, e Pascal um pouco antes, procuraram construir uma máquina de calcular. Nota-se portanto que o mesmo impulso intelectual que o levou ao desenvolvimento da Lógica Matemática o conduziu à busca da mecanização dos processos de raciocínio.

Interessa também chamar a atenção sobre a idéia de uma linguagem artificial que já aparece em Leibniz. Como já foi dito, ele captou muito bem as inúmeras ambigüidades a que estão submetidas as linguagens de comunicação ordinárias e as vantagens que apresentam os símbolos (que ele chamava *notae*) da Aritmética e Álgebra, ciências nas quais a dedução consiste no emprego de caracteres. Ao querer dar à Lógica uma linguagem livre de ambigüidades e ao procurar associar a cada idéia um sinal e obter a solução de todos os problemas mediante a combinação desses sinais, Leibniz acabou provocando um novo desenvolvimento da própria lógica.

Figura 11: Máquinas calculadoras de Leibniz e Pascal

A idéia de uma linguagem artificial ou a redução do raciocínio ao cálculo, como já visto em Lúlio e agora em Leibniz, não é, portanto, patrimônio do século XX. A contribuição de Leibniz ao desenvolvimento da lógica aparece sob dois aspectos: ele aplicou com sucesso métodos matemáticos para a interpretação dos silogismos aristotélicos, e apontou aquelas partes da Álgebra que estão abertas a uma interpretação não aritmética **([Bri79a], volume XI)**. Pela primeira vez se expôs de uma maneira clara o princípio do procedimento formal. Leibniz tornou-se assim o grande precursor da Lógica Matemática.

Talvez se pudesse perguntar como é possível que muitos apresentem a Lógica Simbólica como fruto do nosso tempo, enquanto teve sua origem na segunda metade do

século XVII. É que, na realidade, a contribuição de Leibniz ficou substancialmente reduzida a um mero programa, do qual só executou alguns fragmentos, muito parciais se bem que muito interessantes também, capazes de nos dar uma idéia de como concebia sua obra. Nem sequer seus seguidores diretos levaram para a frente a construção do cálculo lógico mais além de um nível muito rudimentar. Provavelmente a excessiva magnitude do plano de sua *characteritica universalis* o tenha seduzido, afastando Leibniz de objetivos mais modestos porém alcançáveis, como o de construir o primeiro cálculo lógico autêntico.

Ainda dentro desses primeiros passos mais concretos em direção à construção de um dispositivo para cálculo automático, não se pode deixar de falar do ilustre francês Blaise Pascal (1623-1662), já acima citado, que foi matemático, físico, filósofo e brilhante escritor de religião, fundador da teoria moderna das probabilidades. Aos 17 anos já publicava ensaios de matemática que impressionaram a comunidade do seu tempo. Antecedendo a Leibniz, montou a primeira máquina de cálculo digital para ajudar nos negócios do pai em 1642. Iria produzir ainda outras 49 máquinas a partir deste primeiro modelo. Estudos posteriores em geometria, hidrodinâmica, hidrostática e pressão atmosférica o levaram a inventar a seringa e prensa hidráulica e descobrir as famosas Leis de Pressão de Pascal. Após intensa experiência mística, entrou em 1654 para o convento de Port-Royal, onde escreveu pequenos opúsculos místicos. Os últimos anos de sua vida foram dedicados à pesquisa científica **[Wil97]**.

4.2.2 O problema da notação

> *O símbolo não é uma mera formalidade, é a verdadeira essência da álgebra. Sem o símbolo, o objeto é somente uma percepção humana e reflete todas as fases sob as quais os sentidos humanos o captam; substituído por um símbolo o objeto torna-se uma completa abstração, um mero* operando *sujeito a determinadas operações específicas.*
>
> *Tobias Dantzig*

Nas ciências, as descobertas que podem ser compreendidas e assimiladas rapidamente por outros são fonte de progresso imediato. E na Matemática, o conceito de notação está relacionado com isso. Basta lembrar os algarismos romanos e pensar na complexidade que envolve, por exemplo a multiplicação, de MLXXXIV por MMLLLXIX.

Há uma forte analogia entre a história da Álgebra e a da Aritmética. No caso desta última, os homens esforçaram-se durante centenas de anos usando uma numeração inadequada, devido à ausência de um símbolo para o 'nada' (zero). Na Álgebra, a ausência de uma notação reduziu-a a uma coleção de regras fortuitas para a solução de equações numéricas. A descoberta do zero criou a Aritmética que é hoje ensinada e utilizada, e o mesmo se pode dizer em relação à notação, que acabou por introduzir uma nova etapa na história da Álgebra.

As letras liberaram a Álgebra da dependência do uso de palavras, sujeitas às ambigüidades e diferentes interpretações a que está sujeito o discurso de uma linguagem natural (português, inglês, e outras). Mais ainda: a letra ficou livre dos tabus que a associavam

à palavra. Ela é agora um símbolo cujo significado pode transcender o objeto simbolizado. O 'x' de uma equação tem existência própria, independente do objeto que represente.

Importante também é o fato de se poder operar com letras e transformar expressões algébricas com literais, mudando um sentença qualquer para diferentes formas com sentido equivalente. A Álgebra não se torna somente uma maneira mais econômica de se escrever, mas generaliza procedimentos. Por exemplo: expressões tais como 2x + 3, 3x – 5, $x^2 + 4x +7$ tinham antes uma individualidade própria, eram fechadas em si mesmas através das palavras com que eram expressas. Sua resolução dependia de uma apropriada interpretação e manipulação. Com a notação através de literais é possível passar de um individual para um coletivo. A forma linear $ax+b$, a forma quadrática ax^2+bx+c são agora espécies, 'moldes' específicos, e graças a isso foi possível o nascimento da teoria geral das funções que é a base de toda matemática aplicada **[Dan54]**.

A notação de Leibniz usada para o cálculo contribuiu mais do que a de Newton para a difusão das novas idéias sobre integrais*, na época. Pense-se por um momento como se resolve $ax = b$. Imediatamente pode ser dado como resposta que $x = b/a$ e haveria surpresa se alguém respondesse $a = b/x$. É que normalmente se usam as últimas letras do alfabeto para representar as incógnitas e as do começo para representar as quantidades conhecidas. Mas isso não foi sempre assim, e somente no século XVII, a partir de Viète e com Descartes, tais convenções começaram a ser usadas **[Boy74]**.

Geralmente tende-se a apreciar o passado desde o sofisticado posto de observação do tempo atual. É necessário valorizar e revalorizar este difícil e longo passado de pequenas e grandes descobertas. Leibniz, em seu esforço no sentido de reduzir as discussões lógicas a uma forma sistemática que pudesse ser universal, aproximou-se da Lógica Simbólica formal: símbolos ou ideogramas deveriam ser introduzidos para representar um pequeno número de conceitos fundamentais necessários ao pensamento. As idéias compostas deveriam ser formadas a partir desses "alfabetos" do pensamento humano, do mesmo modo como as fórmulas são desenvolvidas na Matemática **[Boy74]**. Isso o levou, entre outras coisas, a pensar em um sistema binário para a Aritmética e a demonstrar a vantagem de tal sistema sobre o decimal para dispositivos mecânicos de calcular†. A idéia de uma lógica estritamente formal – da construção de sistemas sem significado smântico, interpretáveis a posteriori – não tinha surgido. Duzentos anos mais tarde, George Boole formularia as regras básicas de um sistema simbólico para a lógica matemática, refinado posteriormente por outros matemáticos e aplicado à teoria dos conjuntos **([Bri79a], volume III)**. A *álgebra booleana* constituiu a base para o projeto de circuitos usados nos computadores eletrônicos digitais.

4.3 A lógica matemática no século XIX

* Newton e Leibniz descobriram o princípio fundamental do cálculo de que uma integração pode ser feita mais facilmente pela inversão do processo de diferenciação, no cálculo das áreas**[RA91]**.

† Em 1673 Gottfried Leibniz, usando uma engrenagem dentada, construiu uma calculadora capaz de multiplicar, na qual um número era repetido e automaticamente somado a um acumulador.

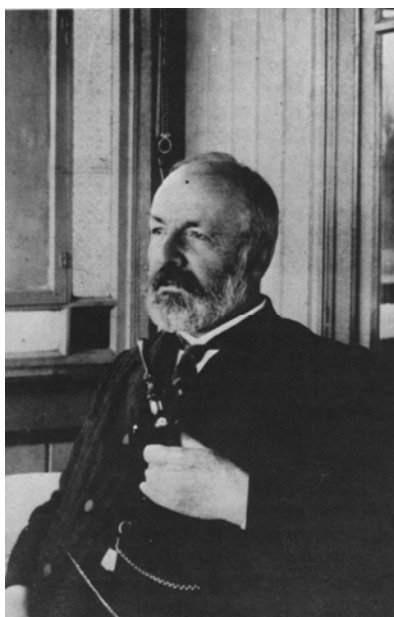



Figura 12: George Cantor

A passagem do século XVIII para o século XIX marca o início de um novo tempo na História da matemática, e, mais do que qualquer período precedente, mereceu ser conhecido como a sua "Idade áurea", e que afetará diretamente a evolução em direção ao surgimento e fundamentação da Ciência da Computação. O que se acrescentou ao corpo da Matemática durante esses cem anos, supera de longe tanto em quantidade como em qualidade a produção total combinada de todas as épocas precedentes. Com uma possível exceção do período conhecido como "Idade heróica", na Grécia antiga, foi uma das mais revolucionárias etapas do desenvolvimento dessa ciência, e, conseqüentemente, também da Computação. Será particularmente objeto de estudo a evolução da Lógica Simbólica – ou Lógica Matemática – que teve Leibniz como predecessor distante.

A partir de meados do século XIX, a *lógica formal** se elabora como um cálculo algébrico, adotando um simbolismo peculiar para as diversas operações lógicas. Graças a esse novo método, puderam-se construir grandes sistemas axiomáticos de lógica, de maneira parecida com a matemática, com os quais se podem efetuar com rapidez e simplicidade raciocínios que a mente humana não consegue espontaneamente.

A Lógica Simbólica – Lógica Matemática a partir daqui –, tem o mesmo objeto que a lógica formal tradicional: estudar e fazer explícitas as formas de inferência, deixando de lado – por abstração – o conteúdo de verdades que estas formas possam transmitir **[San82]**. Não se trata aqui de estudar Lógica, mas de chamar a atenção para a perspectiva que se estava abrindo com o cálculo simbólico: a automatização de algumas operações do pensamento. A Máquina de Turing, como será visto, conceito abstrato que efetivamente deu início à era dos

* A Lógica Formal remonta particularmente a Aristóteles, como visto no capítulo dos *Primórdios*, que a fundiu com a lógica filosófica em um conjunto de obras que posteriormente chamou-se *Organon*. A lógica formal analisa detalhadamente as diversas formas que podem adotar as operações lógicas, em particular o raciocínio, com uma relativa independência dos seus conteúdos concretos.

computadores, baseou-se no princípio de que a simples aplicação de regras permite passar mecanicamente de uns símbolos a outros, sistema lógico que foi inaugurado pelo matemático George Boole.

Entretanto a lógica booleana estava limitada ao raciocínio proposicional, e somente mais tarde, com o desenvolvimento dos quantificadores, a lógica formal estava pronta para ser aplicada ao raciocínio matemático em geral. Os primeiros sistemas foram desenvolvidos por F.L.G. Frege e G. Peano. Ao lado destes, será necessário citar George Cantor (1829 - 1920), matemático alemão que abriu um novo campo dentro do mundo da análise, nascida com Newton e Leibniz, com sua teoria sobre conjuntos infinitos **[Bel37]**.

No início do século XX a Lógica Simbólica se organizará com mais autonomia em relação à matemática e se elaborará em sistemas axiomáticos desenvolvidos, que se colocam em alguns casos como fundamento da própria matemática e que prepararão o surgimento do computador.

4.3.1 Boole e os fundamentos da Lógica Matemática e da Computação

O inglês George Boole (1815 - 1864) é considerado o fundador da Lógica Simbólica **([Bri79a], volume III)**. Ele desenvolveu com sucesso o primeiro sistema formal para raciocínio lógico. Mais ainda, Boole foi o primeiro a enfatizar a possibilidade de se aplicar o cálculo formal a diferentes situações, e fazer operações com regras formais, desconsiderando noções primitivas.

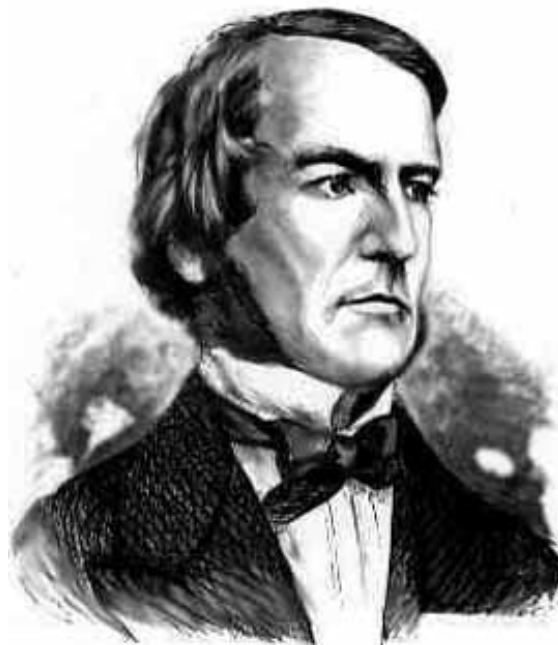



Figura 13: George Boole

Sem Boole, que era um pobre professor autodidata em Matemática, o caminho pelo qual se ligou a Lógica à Matemática talvez demorasse muito a ser construído. Com relação à Computação, se a Máquina Analítica de Babbage (ver capítulo sobre a *Pré-História Tecnológica*)

foi apenas uma tentativa bem inspirada (que teve pouco efeito sobre os futuros construtores do computador), sem a álgebra booleana, no entanto, a tecnologia computacional não teria progredido com facilidade até a velocidade da eletrônica.

Durante quase mais de dois mil anos, a lógica formal dos gregos, conhecida pela sua formulação silogística, foi universalmente considerada como completa e incapaz de sofrer uma melhora essencial. Mais do que isso, a lógica aristotélica parecia estar destinada a ficar nas fronteiras da metafísica, já que somente se tratava, a grosso modo, de uma manipulação de palavras. Não se havia ainda dado o salto para um simbolismo efetivo, embora Leibniz já tivesse aberto o caminho com suas idéias sobre o "alfabeto do pensamento"*.

Foi Boole, em sua obra *The Mathematical Analysis of Logic* (1847), quem forneceu uma idéia clara de formalismo, desenvolvendo-a de modo exemplar. Ele percebeu que poderia ser construída uma álgebra de objetos que não fossem números, no sentido vulgar, e que tal álgebra, sob a forma de um cálculo abstrato, seria capaz de ter várias interpretações **[Kne68]**. O que chamou a atenção na obra foi a clara descrição do que seria a essência do cálculo, isto é, o formalismo, procedimento, conforme descreveu, "cuja validade não depende da interpretação dos símbolos mas sim da exclusiva combinação dos mesmos" **[Boc66]**. Ele concebeu a lógica como uma construção formal à qual se busca posteriormente uma interpretação.

Boole criou o primeiro sistema bem sucedido para o raciocínio lógico, tendo sido pioneiro ao enfatizar a possibilidade de se aplicar o cálculo formal em diferentes situações e fazer cálculos de acordo com regras formais, desconsiderando as interpretações dos símbolos usados. Através de símbolos e operações específicas, as proposições lógicas poderiam ser reduzidas a equações e as equações silogísticas poderiam ser computadas de acordo com as regras da álgebra ordinária. Pela aplicação de operações matemáticas puras e contando com o conhecimento da álgebra booleana é possível tirar qualquer conclusão que esteja contida logicamente em qualquer conjunto de premissas específicas.

De especial interesse para a Computação, sua idéia de um sistema matemático baseado em duas quantidades, o 'Universo' e o 'Nada', representados por '1' e '0', o levou a inventar um sistema de dois estados para a quantificação lógica. Mais tarde os construtores do primeiro computador entenderam que um sistema com somente dois valores pode compor mecanismos para perfazer cálculos†.

George Boole estava convencido de que sua álgebra não somente tinha demonstrado a equivalência entre Matemática e Lógica, como representava a sistematização do pensamento humano. Na verdade a ciência, depois dele, principalmente com Husserl, pai da Fenomenologia, demonstrará que a razão humana é mais complicada e ambígua, difícil de ser conceituada e mais poderosa que a lógica formal, mas do ponto de vista da Matemática e da Computação, a álgebra booleana foi importante, e só os anos fizeram ver, pois a lógica até

---

* Leibniz já tinha compreendido no século XVII que há alguma semelhança entre a disjunção e conjunção de conceitos e a adição e multiplicação de números mas foi difícil para ele formular precisamente em que consistia essa semelhança e como usá-la depois como base para um cálculo lógico **[Kne68]**.

† A base do *hardware* sobre a qual são construídos todos os computadores digitais é formada de dispositivos eletrônicos diminutos denominados **portas lógicas**. É um circuito digital no qual somente dois valores lógicos estão presentes. Para se descrever os circuitos que podem ser construídos pela combinação dessas portas lógicas é necessária a álgebra booleana.

então era incompleta e não explicava muitos princípios de dedução empregados em raciocínios matemáticos elementares.

Mas, a lógica booleana estava limitada ao raciocínio proposicional, e somente após o desenvolvimento de quantificadores, introduzidos por Peirce, é que a lógica formal pôde ser aplicada ao raciocínio matemático geral. Além de Peirce, também Schröeder e Jevons* aperfeiçoaram e superaram algumas restrições do sistema booleano: disjunção exclusiva, emprego da letra *v* para exprimir proposições existenciais, admissão de coeficientes numéricos além do 0 e 1 e o emprego do sinal de divisão. O resultado mais importante, no entanto, foi a apresentação do cálculo de uma forma extremamente axiomatizada.

4.3.2 A importância de Frege e Peano

Frege (1848 - 1925) e Peano (1858 - 1932) trabalharam para fornecer bases mais sólidas à álgebra e generalizar o raciocínio matemático **[Har96]**.

Gottlob Frege ocupa um lugar de destaque dentro da Lógica. Embora não tão conhecido em seu tempo e bastante incompreendido, deve-se ressaltar que ainda hoje torna-se difícil descrever a quantidade de conceitos e inovações, muitos revolucionários, que elaborou de forma exemplar pela sua sistematização e clareza. Muitos autores comparam seu *Begriffsschrift* aos Primeiros Analíticos de Aristóteles, pelos pontos de vista totalmente geniais.

Frege foi o primeiro a formular com precisão a diferença entre *variável* e *constante*, assim como o conceito de *função lógica,* a idéia de uma *função de vários argumentos* e o conceito de *quantificador*†. A ele se deve uma conceituação muito mais exata da teoria aristotélica sobre sistema axiomático, assim como uma clara distinção entre lei e regra, linguagem e metalinguagem. Ele é autor da teoria da descrição e quem elaborou sistematicamente o conceito de valor. Mas isto não é tudo, pois todas estas coisas são apenas produtos de um empreendimento muito maior e fundamental, que o inspirou desde suas primeiras pesquisas: uma investigação das características daquilo que o homem diz quando transmite informação por meio de juízos.

Na verdade o que Frege chamou de Lógica – assim como seus contemporâneos Russell e Wittgenstein – não é o que hoje é chamado Lógica, fruto do formalismo e da teoria dos conjuntos que acabaram por predominar entre os matemáticos, mas sim o que se denomina *semântica*, uma disciplina sobre o conteúdo, natureza desse conteúdo e estrutura. Ele gastou considerável esforço na separação de suas concepções lógicas daquelas concepções dos 'lógicos computacionais' como Boole, Jevons e Schröeder. Estes estavam, como já foi dito, empenhados no desenvolvimento de um cálculo do raciocínio como Leibniz propusera, mas Frege queria algo mais ambicioso: projetar uma *língua characteristica*. Dizia ele que uma das

---

* Jevons foi o primeiro a compreender que os métodos booleanos podem ser reduzidos às regras do cálculo elementar, com a possibilidade, portanto, de ser mecanizados. Em 1869 conseguiu construir uma máquina lógica apresentada no ano seguinte ao público: era um dispositivo de 21 chaves para testar a validade de inferências na lógica equacional. Algumas das características deste dispositivo foram usadas mais tarde na implementação do computador. A máquina está conservada no museu de História da Ciência em Oxford.

† O emprego de quantificadores para ligar variáveis, principal característica do simbolismo lógico moderno e que o torna superior em alguns aspectos à linguagem vulgar e ao simbolismo algébrico de Boole, está entre as maiores invenções intelectuais do século XIX **[Kne68]**.

tarefas da filosofia era romper o domínio da palavra sobre o espírito humano. Frege procurou usar um sistema simbólico, que até então somente se pensava para a matemática, também para a filosofia: um simbolismo que retratasse o que se pode dizer sobre as coisas. Ele buscava algo que não somente descrevesse ou fosse referido a coisas pensadas, mas o próprio pensar **[Cof91]**.

Os lógicos tradicionais estavam basicamente interessados na solução de problemas tradicionais de lógica, como por exemplo a validade. O objetivo de Frege foi mais além: entrou no campo da semântica, do conteúdo, do significado, onde encontrou o fundamento último da inferência, da validade, etc. Frege acabou derivando para uma filosofia da lógica e da matemática e influenciou diretamente a Russell, David Hilbert, Alonzo Church e Carnap. Destes, Hilbert e Church têm um papel decisivo na História conceitual da Ciência da Computação.

Frege desejava provar que não somente o raciocínio usado na matemática, mas também os princípios subjacentes – ou seja, toda a matemática – são pura lógica. Porém ele expressou suas buscas e resultados – pelos quais acabou sendo considerado um dos pais da Lógica moderna, de uma forma excessivamente filosófica, em uma notação matemática não convencional. O mérito maior dele foi elaborar uma concepção lógica mais abrangente do que a Lógica de Aristóteles. Em um procedimento que lembra a "characteristica universalis"*, Frege construiu um sistema especial de símbolos para desenvolver a lógica de maneira exata e foi muito além das proposições e dos argumentos. Em sua grandes obras, *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* (Ideografia, uma linguagem formalizada do pensamento puro construída de modo aritmético) e *Grundgesitze der Arithmetik. Begriffsschriflich abgeleitet* (Leis Fundamentais da Aritmética, Ideograficamente Deduzidas), está contida de modo explícito e plenamente caracterizado uma série de conceitos – conectivos, função, função proposicional, quantificadores, etc. – que seriam vitais para a Lógica Matemática a partir de então **[Gom97]**.

Foi através do contato com a obra de Frege que Bertrand Russell (1872 - 1970) procurou levar avante a idéia de construir toda a matemática sobre bases lógicas, convencido de que ambas são idênticas. Os postulados fregianos†, adotados primeiramente por Peano, foram incorporados por Russell, que estendeu as teses logicistas de Frege à Geometria e às disciplinas matemáticas em geral.

Peano tinha objetivo semelhante a Frege, mas mais realista. Ele desenvolveu uma notação formal para raciocínio matemático que procurasse conter não só a lógica matemática, mas todos os ramos mais importantes dela. O simbolismo de Peano e seus axiomas – dos quais dependem muitas construções rigorosas na álgebra e na análise – "representam a mais notável tentativa do século XIX de reduzir a aritmética comum, e portanto a maior parte da matemática, a um puro simbolismo formal. Aqui o método postulacional atingiu novo nível de precisão, sem ambigüidade de sentido, sem hipóteses ocultas" **[Boy74]**. Para maiores detalhes, ver anexo sobre *A Aritmética de Peano*.

---

* Como já foi dito, a idéia lançada por Leibniz de uma linguagem filosófica que seria um simbolismo através do qual o homem estaria em condições de expressar seus pensamentos com plena clareza e dirimir dúvidas através de simples cálculos.

† Sobre número, dedução, inferência, proposições, premissas, etc.

Já Hilbert procurou colocar em prática a teoria da demonstração de Frege, e pode-se ver nessas palavras deste as idéias implementadas posteriormente no programa hilbertiano: "a inferência procede, pois, em meu sistema de escrita conceitual (Begriffsschrift), seguindo uma espécie de cálculo. Não me refiro a este em sentido estrito, como se fosse um algoritmo que nele predominasse, (...), mas no sentido de que existe um algoritmo total, quer dizer, um conjunto de regras que resolvem a passagem de uma proposição ou de duas, a outra nova, de tal forma que nada se dá que não esteja de acordo com estas regras. Minha meta é, pois, uma ininterrupta exigência de precisão no processo de demonstração, e a máxima exatidão lógica, ao mesmo tempo que clareza e brevidade" **[Boc66]**.

Pode-se notar a partir desse momento uma guinada no conceito de Lógica: os objetos da investigação lógica já não são mais as próprias fórmulas, mas as regras de operação pelas quais se formam e se deduzem.

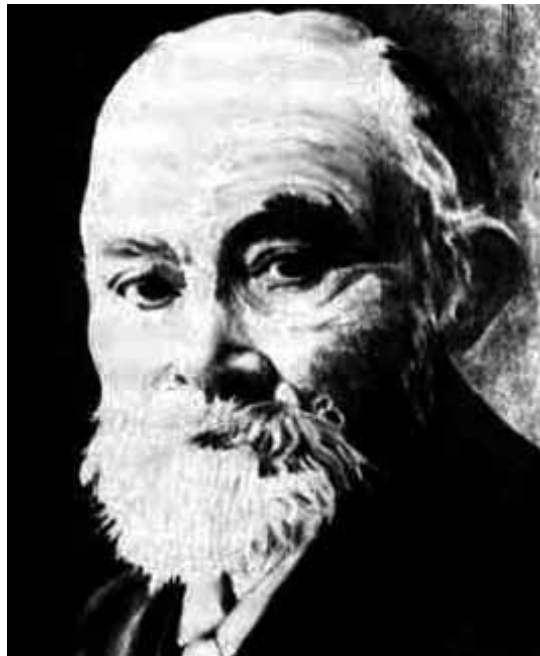



Figura 14: Frege

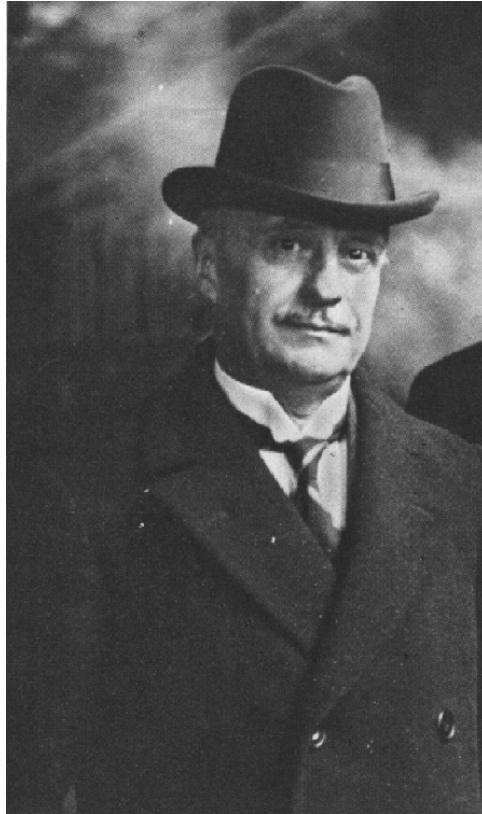



Figura 15: Peano

4.4 O desenvolvimento da Lógica Matemática

Uma das metas dos matemáticos no final do século XIX foi a de obter um *rigor conceitual* das noções do cálculo infinitesimal (limite, continuidade, infinito matemático, etc.). Tal programa foi chamado de "aritmetização da análise", isto é, a busca da redução dos conceitos fundamentais da análise (a matemática que tem como base a teoria dos números reais) aos conceitos da aritmética (a matemática que tem como base a teoria dos número inteiros positivos, isto é, dos números naturais e, por extensão, dos números racionais).

Por exemplo, ao invés de se tomar o número imaginário $\sqrt{-1}$ como uma entidade um tanto misteriosa, pode-se defini-lo como um par ordenado de números inteiros (0,1), sobre o qual se realizam certas operações de "adição" e "multiplicação". Analogamente, o número irracional $\sqrt{2}$ se definia numa certa classe de números irracionais, cujo quadrado é menor do que 2. Dado que a Geometria podia ser reduzida à Análise (Geometria Analítica), a Aritmética vinha a se configurar como a base natural de todo o edifício matemático. O ponto culminante deste processo foram os axiomas de Peano (1899), que fundamentaram toda a Aritmética elementar posterior.

Ao mesmo tempo, matemáticos como Frege, Cantor e Russell, não convencidos da "naturalidade" da base constituída pela aritmética. Eles procuravam conduzir a própria aritmética a uma base mais profunda, reduzindo o conceito de número natural ao conceito lógico de classe, ou, para recorrer a Cantor, definir número em termos de conjunto. Deste modo, a lógica das classes apresentava-se como a teoria mais adequada para a investigação

sobre os fundamentos da matemática. O esforço dos matemáticos foi o de dar à álgebra uma estrutura lógica, procurando caracterizar a matemática não tanto pelo seu conteúdo quanto pela sua forma.

Bochenski **[Boc66]**, falando da história da Lógica Matemática, diz que a partir de 1904, com Hilbert, inicia-se um novo período dessa ciência então emergente, que se caracteriza pela aparição da Metalógica* (Hilbert, Löwenheim e Scholem) e, a partir de 1930, por uma sistematização formalista dessa mesma Metalógica. Iniciaram-se discussões sobre o valor e os limites da axiomatização, o nexo entre Lógica e Matemática, o problema da verdade (Hilbert, Gödel, Tarski).

A Metalógica, em sua vertente *sintática* ocupa-se das propriedades externas dos cálculos, como por exemplo, a consistência, a completude, a decidibilidade dos sistemas axiomáticos e a independência dos axiomas. Hilbert, Gödel e Church são autores nesse campo. Em sua parte *semântica*, a Metalógica dirige-se ao significado dos símbolos, dos cálculos com relação a um determinado mundo de objetos. Tarski, Carnap e Quino, entre outros se interessaram por estas questões.

Apareceram também novos sistemas lógicos: as *lógicas naturais*, de Gentzen e Jaskowski, *lógica polivalente* de Post e Lukasiewicz, e a *lógica intuicionista* de Heytings.

Complementando essas idéias cabe destacar alguns sistemas originais de outros matemáticos como Schönfinkel (1924), Curry (1930), Kleene (1934), Rosser (1935) e o já citado Alonzo Church (1941). Deve-se lembrar que quase todos estes últimos, junto com o logicista inglês Alan M. Turing, acabaram por definir, antes mesmo de existir o computador propriamente, a *natureza da computação*, e as implicações e limites do pensamento humano através de uma máquina.

4.5 A crise dos fundamentos e as tentativas de superação

Os matemáticos são conhecidos por serem exigentes na hora de pedir uma prova absoluta antes de aceitarem qualquer afirmação. Esta reputação é claramente mostrada em uma anedota:

> Um astrônomo, um físico e um matemático estavam passando férias na Escócia. Olhando pela janela do trem eles avistaram uma ovelha preta no meio de um campo. "Que interessante", observou o astrônomo, "na Escócia todas as ovelhas são pretas." Ao que o físico respondeu: "Não, nada disso!. Algumas ovelhas escocesas são pretas." O matemático olhou para cima em desespero e disse: "Na Escócia existe pelo menos um campo, contendo pelo menos uma ovelha e pelo menos um lado dela é preto." (Ian Stuart, Conceitos de matemática moderna, in **[Sin99]**)

* Quando a própria Lógica Formal reflete sobre seus conteúdos.

E o matemático que se especializa no estudo da lógica matemática é ainda mais rigoroso do que o matemático comum. Os matemáticos lógicos começaram a questionar idéias que os outros matemáticos consideravam certas há séculos. Por exempo, a lei da *tricotomia* declara que cada número é ou negativo, ou positivo ou então zero. Ninguém se preocupara em provar isso que parece óbvio, mas os lógicos perceberam que se não o fosse, ela poderia ser falsa, e todo o edifício matemático que dependia dessa lei desmoronaria*. Apesar das disciplinas dedutivas terem atingido um alto grau de perfeição lógica, algumas dúvidas começaram a abalar a confiança dos matemáticos: o surgimento, por volta de 1900, de inúmeros paradoxos ou antinomias, especialmente na teoria dos conjuntos†. O aparecimento de tais contradições mostrava que havia algum defeito nos métodos. Será que se poderia ter certeza de que em se usando os axiomas de um sistema rigidamente lógico – o grande sonho de tantos matemáticos do início do século XX de reduzir a matemática e o conhecimento à lógica –, nunca se chegaria a uma contradição, dentro dos axiomas do sistema? Estava iminente, nos fins do século XIX, uma inevitável colisão entre matemática e filosofia. Alguns vagos conceitos metafísicos associados com o pensamento humano já tinham chamado a atenção de matemáticos das duas primeiras décadas do século XX‡, que passaram a procurar a verdadeira natureza do raciocínio dentro da ciência matemática. O que é um procedimento correto, qual a relação entre verdade e demonstração, é possível fornecer uma prova para todos os enunciados matemáticos verdadeiros? E o problema das ambigüidades, já que a matemática sempre foi feita através de uma linguagem natural?

Desde os antigos gregos a matemática vem acumulando mais teoremas e verdades, e, embora a maioria deles tenha sido rigorosamente provada, os matemáticos temiam que alguns casos tivessem sido aceitos sem um exame mais adequado. Os lógicos então decidiram provar todos os teoremas, a partir de princípios fundamentais. No entanto, cada verdade tinha sido deduzida de outras verdades. E estas, por sua vez, de outras ainda mais fundamentais e assim por diante. Os lógicos acabaram por chegar aos axiomas da matemática, essas declarações essenciais tão fundamentais que não podiam ser provadas, pois são auto-evidentes§. O desafio para os lógicos era reconstruir toda a matemática a partir desses axiomas.

Uma legião de lógicos participou deste processo, lento e doloroso, usando somente um número mínimo de axiomas. A idéia era consolidar, através do raciocínio lógico e rigoroso, o que já se pensava ser conhecido. Este quadro estimulou a criatividade matemática.

---

* A lei da tricotomia foi demonstrada no final do século XIX.

† O paradoxo de Russel, o paradoxo de Cantor, o paradoxo de Burati Forti, o paradoxo de Richard, etc. Para exemplificar, vamos ao de Cantor, descoberto por ele próprio em 1899: se S é o conjunto de todos os conjuntos, então seus subconjuntos devem estar também entre os seus elementos. Conseqüentemente, o número cardinal de S não pode ser menor do que o do conjunto dos subconjuntos de S. Mas isso, pelo teorema do próprio Cantor, deveria ocorrer!

‡ Basta ler as palavras do matemático Sylvester em sua controvérsia com Huxley. Dizia que a matemática se origina "diretamente das forças e atividades inerentes da mente humana, e da introspecção continuamente renovada daquele mundo interior do pensamento em que os fenômenos são tão variados e exigem atenção tão grande quanto os do mundo físico exterior". Para ele, a matemática era "revelar as leis da inteligência humana", assim como a física revela as leis do mundo dos sentidos **[Cos77]**.

§ A lei comutativa da adição, por exemplo: *para quaisquer números a e b,* a + b = b + a.

Na tentativa de se resolverem os paradoxos surgiram três grandes escolas da lógica: a Logicista* , a Intuicionista† e a Formalista.

A escola logicista rapidamente ficou exposta a fortes críticas‡. Frege, Peano e Russell , devido ao seu platonismo, acreditavam em um mundo objetivo, existente por si mesmo, de entes e relações matemáticas que o pesquisador deve descobrir e não inventar. Bertrand Russell tinha objetivos ainda maiores: utilizar o instrumental da lógica como ponto de partida do pensamento filosófico, através da geração de uma linguagem perfeita. Mas a matemática, enquanto perquirição pura, independe teoricamente dessas aplicações, bastando ver as pesquisas atuais. Deve-se, no entanto, destacar o mérito dessa escola de incrementar grandemente o progresso da logística e confirmar a união íntima entre matemática e lógica.

O programa intuicionista sofreu também fortes críticas, principalmente a de desfigurar a matemática, tornando-a algo subjetivo e praticamente impossível. O próprio modo de se provar a não-contradição de uma teoria matemática, buscando um 'modelo' dos axiomas dessa teoria dentro de outra teoria já existente (e que era considerada coerente§) mostrou-se pouco confiável: como dar a certeza da não-contraditoriedade dessa outra teoria? A maior parte dos matemáticos dos nossos dias afastou-se dessa linha de pensamento. Positivamente falando, sua dura crítica à matemática tradicional obrigou os especialistas nos fundamentos a desenvolverem novos métodos para reabilitar a teoria clássica. A escola formalista progrediu bastante através das polêmicas com os intuicionistas **[Cos77]**.

4.5.1 A figura de David Hilbert

Para David Hilbert (1862 - 1943) e outros, o problema de estabelecer fundamentos rigorosos era o grande desafio ao empreendimento de tantos, que pretendiam reduzir todas as leis científicas a equações matemáticas. Ele acreditava que tudo na matemática poderia e deveria ser provado a partir dos axiomas básicos. O resultado disso levaria a demonstrar conclusivamente, segundo ele, os dois elementos mais importantes do sistema matemático. Em primeiro lugar a matemática deveria, pelo menos em teoria, ser capaz de responder a cada pergunta individual – este é o mesmo espírito de completude que no passado exigira a invenção de números novos, como os negativos e os imaginários. Em segundo lugar, deveria ficar livre de inconsistências – ou seja, tendo-se mostrado que uma declaração é verdadeira por um método, não deveria ser possível mostrar que ela é falsa por outro método. Hilbert estava convencido de que, tomando apenas alguns axiomas, seria possível responder a qualquer pergunta matemática imaginária, sem medo de contradição. O esforço para

---

* A tese logiscista compõe-se de duas partes: 1)Toda idéia matemática pode ser definida por intermédio de conectivos lógicos (classe ou conjunto, implicação, etc.); 2)Todo enunciado matematicamente verdadeiro pode ser demonstrado a partir de princípios lógicos ("não contradição", "terceiro excluído", etc.), mediante raciocínios puramente matemáticos.

† Para Brower, fundador desta escola – na verdade um radicalizador das teses de Kronecker que não aceitava a teoria dos conjuntos – o saber matemático escapa a toda e qualquer caracterização simbólica e se forma em etapas sucessivas que não podem ser conhecidas de antemão: a atividade do intelecto cria e dá forma a entes matemáticos, aproximando-se do apriorismo temporal de Kant.

‡ Os logicistas tiveram de apelar a princípios extra-lógicos – axioma de Zermelo, axioma do infinito – que ainda hoje encontram-se sujeitos a calorosos debates e fortes reparos.

§ Caminho praticado por Hilbert no seu famoso trabalho *Fundamentos da Geometria* (1899), onde axiomatizou de modo rigoroso a geometria euclidiana.

reconstruir logicamente o conhecimento matemático acabou sendo liderado por essa figura, talvez a mais eminente da época.

No dia 8 de agosto de 1900, Hilbert deu uma palestra histórica no Congresso Internacional de Matemática em Paris. Ele apresentou 23 problemas não-resolvidos da matemática que acreditava serem de imediata importância. Alguns problemas relacionavam-se com áreas mais gerais da matemática, mas a maioria deles estava ligada aos fundamentos lógicos dessa ciência. Tais problemas deveriam focalizar a atenção do mundo matemático e fornecer um programa de pesquisas. Hilbert queria unir a comunidade para ajudá-lo a realizar sua visão de um sistema matemático livre de dúvidas ou inconsistências**[Sin99]**. Todos esses estudos denominaram-se *Metamatemática* ou *Metalógica*, pela conectividade das duas.

Ele propôs-se a demonstrar a coerência da aritmética[*], para depois estender tal coerência aos âmbitos dos demais sistemas. Apostou na possibilidade da criação de uma linguagem puramente sintática, *sem significado*, a partir da qual se poderia falar a respeito da verdade ou falsidade dos enunciados. Tal linguagem foi e é chamada de *sistema formal*, e está resumida no anexo *A concepção formalista da matemática*. Isto, que fazia parte do centro da doutrina formalista, mais tarde estimularia Turing a fazer descobertas importantes sobre as capacidades das máquinas. Registre-se também que John von Neumann, a quem muitos atribuem a construção do primeiro computador, era um aluno de Hilbert e um dos principais teóricos da escola formalista[†].

---

[*] No início do século XX a matemática estava reduzida a 3 grandes sistemas axiomáticos: aritmética, análise e conjunto, sendo o mais fundamental o primeiro. Era natural que ele escolhesse esse sistema.

[†] John von Neumann falava 5 línguas e foi um brilhante logicista, matemático e físico. Além de lhe ser atribuída a invenção do primeiro computador, ele estava no centro do grupo que criou o conceito de "programa armazenado", que potencializou extremamente o poder computacional das máquinas que então surgiam.

Figura 16: Professor David Hilbert

Interessam agora dois problemas da referida lista de 23. O segundo, relacionado com a confiabilidade do raciocínio matemático (isto é, se ao seguir as regras de determinado raciocínio matemático não se chegaria a contradições), e, ligado a ele, o problema de número dez. Este era de enunciado bastante simples: descreva um algoritmo que determine se uma dada equação diofantina do tipo $P(u_1,u_2,...,u_n) = 0$, onde P é um polinômio com coeficientes inteiros, tem solução dentro do conjunto dos inteiros. É o famoso problema da *decidibilidade, o Entscheidungsproblem*. Consistia em indagar se existe um procedimento mecânico efetivo para determinar se todos os enunciados matemáticos verdadeiros poderiam ser ou não provados, isto é, se eles poderiam ser deduzidos a partir de um dado conjunto de premissas[*]. Para entender um pouco mais sobre *decisão* na matemática, ver o anexo *O problema da decisão na Matemática*.

Também a questão da consistência, como já foi dito, era decisiva para ele, pois é uma condição necessária para o sistema axiomático do tipo que ele tinha em mente. Aristóteles já tinha mostrado que se um sistema é inconsistente, *qualquer* afirmação poderia ser provada como falsa ou verdadeira. Neste caso não seria possível haver um fundamento sólido

---

[*] A simplicidade do problema de Hilbert é apenas aparente, e somente após 70 anos de esforços foi encontrada a solução, por Matijasevic, um matemático russo de apenas 22 anos na época. É uma solução bastante complexa, dependendo tanto de resultados da Teoria do Números, conhecidos há muitíssimos anos, como do trabalho anterior de três americanos, Martin Davis, Julia Robinson e Hilary Putnan, que por sua vez baseia-se em certos resultados fundamentais sobre lógica e algoritmos descobertos na década de 30 por Kurt Gödel, Alan Turing, Emil Post, Alonso Church e Stephen Kleene. A resposta a esse problema de Hilbert é: tal algoritmo não existe: o décimo problema é *indecidível.*

para qualquer tipo de conhecimento, matemático ou não. Anos mais tarde, em 1928, no Congresso Internacional de Matemáticos, realizado em Bolonha, Itália, Hilbert lançou um novo desafio, que na verdade somente enfatizava aspectos do segundo e do décimo problema já descritos. Hilbert queria saber se é possível provar toda assertiva matemática verdadeira. Estava buscando algo como uma "máquina de gerar enunciados matemáticos verdadeiros": uma vez alimentada com um enunciado matemático, poderia dizer se o enunciado é falso ou verdadeiro **[Cas97]**. É um problema que está relacionado com o citado projeto hilbertiano da busca de um sistema formal *completo* e *consistente*.

Ao mesmo tempo, em 1927, com 22 anos, von Neumann publicou 5 artigos que atingiram fortemente o mundo acadêmico. Três deles consistiam em críticas à física quântica, um outro estabelecia um novo campo de pesquisas chamado *Teoria dos Jogos*, e, finalmente, o que mais impactou o desenvolvimento da Computação: era o estudo do relacionamento entre sistemas formais lógicos e os limites da matemática. Von Neumann demonstrou a necessidade de se provar a consistência da matemática, um passo importante e crítico tendo em vista o estabelecimento das bases teóricas da Computação (embora ninguém tivesse esse horizonte por enquanto).

Já foi citado, no capítulo sobre o *Desenvolvimento da Lógica Matemática,* o desafio dos matemáticos do início do século de *aritmetizar a análise*. Eles estavam de acordo, no que diz respeito às proposições geométricas e outros tipos de afirmações matemáticas, em que poderiam ser reformuladas e reduzidas a afirmações sobre números. Logo, o problema da consistência da matemática estava reduzido à determinação da consistência da aritmética. Hilbert estava interessado em dar uma *teoria da aritmética*, isto é, um sistema formal que fosse finitisticamente descritível*, consistente, completo e suficientemente poderoso para descrever todas as afirmações que possam ser feitas sobre números naturais. O que Hilbert queria em 1928 era que, para uma determinada afirmação matemática, por exemplo, "a soma de dois números ímpares é sempre um número par", houvesse um procedimento que, após um número finito de passos, parasse e indicasse se aquela afirmação poderia ou não ser provada em determinado sistema formal, suficientemente poderoso para abranger a aritmética ordinária **[Cas97]**. Isto está diretamente relacionado com o trabalho de Gödel e Alan Turing.

Pode-se afirmar que, em geral, a lógica matemática prestou naqueles tempos maior atenção à *linguagem científica*, já que seu projeto era o da elaboração de uma linguagem lógica de grande precisão, que fosse boa para tornar transparentes as estruturas lógicas de teorias científicas. Tal projeto encontrou seus limites, tanto na ordem sintática como na ordem semântica (por exemplo, com os célebres teoremas de limitação formal). Este fenômeno levou a uma maior valorização da linguagem ordinária, que, apesar de suas flutuações e imprecisões, encerram uma riqueza lógica que os cálculos formais não conseguem recolher de todo. Dentro da própria matemática – como se verá mais adiante com Gödel – há verdades que não podem ser demonstradas mediante uma *dedução formal*, mas que podem ser demonstradas – o teorema da incompletude de Gödel é uma prova disso – mediante um raciocínio metamatemático informal. A partir desse propósito de construção de uma

* O termo *finitístico* é usado por vários autores. Hilbert quis dizer que tal sistema deveria ser construído com um número finito de axiomas e regras e toda prova dentro do sistema deveria ter um número finito de passos.

linguagem ideal surgiu a filosofia da linguagem (Moore, Wittgenstein, Geach em sua segunda etapa) colocando as questões lógicas sobre nova ótica **[San82]**.

Na verdade, tanto a lógica matemática em sentido estrito como os estudos de semântica e filosofia da linguagem depararam-se com problemas filosóficos que não se resolvem somente dentro de uma perspectiva lógica. Há questões de fundo da lógica matemática que pertencem já a uma filosofia da matemática.

Todos esses desafios abriram uma porta lateral para a Computação e deram origem a um novo e decisivo capítulo na sua História. Da tentativa de resolvê-los ocorreu uma profunda revolução conceitual na Matemática – o Teorema de Gödel – e surgiu o fundamento básico de todo o estudo e desenvolvimento da Computação posterior: a Máquina de Turing.

4.6 Kurt Gödel: muito além da lógica*

Kurt Gödel (1906 – 1978) não desfruta do mesmo prestígio de outros cientistas contemporâneos seus, como Albert Einstein. Possivelmente contribua para isto o fato de que suas descobertas se produziram em um campo, o da lógica matemática, próprio das ciências formais, e não em algum ramo da ciência que tenha influenciado diretamente no conjunto da sociedade.

No entanto, suas grandes contribuições à lógica formal se estendem – segundo seus biógrafos – muito além do seu estrito âmbito formal e abordam questões tão vastas e espinhosas como a natureza da verdade, o conhecimento e a certeza. Pode-se afirmar que, junto com a teoria da relatividade de Einstein e o princípio da incerteza de Heisenberg, o teorema da incompletude de Gödel despertou a ciência moderna de seu "sonho dogmático".

4.6.1 Um pouco de história

Nascido em Brünn (hoje Brno, na República Tcheca), Kurt Friedrich Gödel (ao naturalizar-se norte-americano, em 1948, ele deixou o segundo nome) instala-se em Viena em 1924. Logo se apaixona pela cidade, por sua vida universitária e atmosfera intelectual. Requisita a nacionalidade austríaca e, em 26 de fevereiro de 1929, três dias depois da morte de seu pai, deixa oficialmente de ser tcheco. Apesar do luto, termina sua tese de doutorado, *Sobre a completude do cálculo lógico*. Nessa monografia datilografada de apenas 30 páginas, o jovem matemático, então com 23 anos, expõe diversos resultados extremamente importantes para a lógica. Deduz que todo sistema de axiomas de primeira ordem não-contraditório possui um modelo. Isto é, que existe um conjunto de objetos que verificam os axiomas do sistema.

Existem duas definições de completude:

1) Um sistema de axiomas é completo (como uma caixa de ferramentas bem provida, que permite realizar todos os trabalhos necessários) quando todos os teoremas verdadeiros da teoria em pauta (por exemplo, a aritmética) podem ser deduzidos a partir dele. Esta é a completude semântica.

---

* Parte do texto que se segue, a partir do item *Um pouco de história* vem de maravilhoso artigo publicado na revista Scientific American Brasil, edição Gênios da Ciência Matemática: A vanguarda matemática e os limites da razão

2) Um sistema de axiomas é completo (como uma caixa de ferramentas à qual não se pode acrescentar nada) quando toda tentativa de lhe adicionar um novo axioma (independente dos anteriores) resulta em contradição. Esta é a completude sintática.

Gödel usa o termo completude no primeiro sentido e mostra que toda fórmula lógica verdadeira do cálculo de predicados de primeira ordem é demonstrável a partir dos axiomas clássicos desse mesmo cálculo. Para ele, esse resultado “constitui um complemento teórico ao método usual de demonstração de não-contradição”. O “método usual” consiste em construir um modelo, ou seja, oferecer uma interpretação semântica (por exemplo, geométrica) do sistema considerado: se a teoria admite um modelo, então é não-contraditória. Em sua tese, o jovem demonstra a recíproca de tal propriedade: todo sistema de axiomas de primeira ordem não contraditório admite um modelo[*].

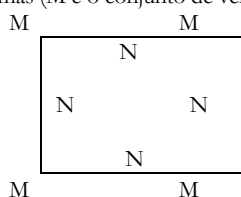
Esse resultado parece confirmar a esperança formalista de David Hilbert, de construir uma teoria rigorosa capaz de descrever toda a matemática. Mas Gödel opta pela prudência. Entusiasmado, seu orientador, Hans Hahn, encoraja-o a publicar uma versão presumida do trabalho. O artigo *Sobre a completude dos axiomas do cálculo lógico de funções* aparece, em 1930, no periódico *Monatshefte für Mathematik und Physik*. Detalhe interessante: no artigo, Gödel não retoma nenhum elemento da introdução de sua tese, na qual havia relacionado seu resultado às discussões acerca dos fundamentos da matemática.

Eis aqui, nas palavras do próprio Gódel, um resumo do artigo: “Na fundamentação axiomática da lógica, tal como é estabelecida, por exemplo, nos *Principia mathematica,* levanta-se a seguinte questão: os axiomas utilizados são de fato completos, ou seja, suficientes para deles se deduzir todas as proposições importantes da lógica pela via formal? Este problema não foi resolvido até agora senão para as proposições lógicas mais simples, especificamente para aquelas do cálculo proposicional. A resposta é positiva, e significa que toda proposição verdadeira (válida em geral) decorre dos axiomas dos *Principia mathematica*. O autor mostra como estender esse teorema às fórmulas do cálculo funcional (cálculo de predicados) de primeira ordem (em que os quantificadores não operam sobre as funções e predicados)”.

Em sua tese, Gödel ainda toma o partido de Hilbert, contra o chamado “intuicionismo” de Luitzen Brouwer (1881-1966). Mas certas idéias do matemático holandês certamente o influenciam. Não sabemos se ele assistiu às duas conferências que Brouwer proferiu em Viena em março de 1928; porém não há dúvida de que conhecia seu conteúdo, ao menos por meio das atas às quais teve acesso. Na primeira dessas duas conferências, intitulada *Matemática, ciência e linguagem,* Brouwer ironiza as tentativas feitas pelos formalistas (Hilbert e

---

[*] Em outras palavras, busca-se uma interpretação semântica para o sistema de axiomas. Ex.: certo sistema de axiomas diz respeito a dois conjuntos M e N cuja natureza, inicialmente, não é especificada. Esses axiomas são: 1) M e N têm o mesmo número de elementos; 2) Nenhum elemento de N contém mais de 2 elementos de M; 3) Nenhum elemento de M está contido em mais de dois elementos de N. Esse sistema é coerente, pois se pode associar a ele o seguinte modelo geométrico, onde se verifica os três axiomas (M é o conjunto de vértices do quadrado e N o conjunto de seus lados):

M M
N
N N
N
M M

seguidores) para tornar mais precisa a linguagem. Segundo ele, não existe na matemática pura, assim como em qualquer outra área, uma linguagem absolutamente segura, isto é, uma linguagem capaz de excluir todo mal-entendido e na qual a memória evite todo erro (por exemplo, a confusão entre entes matemáticos). Como conseqüência, parece-lhe impossível remediar essa situação "ao submeter, como faz a escola formalista, a linguagem matemática a um tratamento matemático" e "pela tentativa de exprimi-Ia em uma linguagem de segunda ordem ou em uma metalinguagem". O desacordo entre Gödel e Brouwer acerca do assunto é apenas uma questão de interpretação: enquanto Brouwer vê como confissão de fraqueza a impossibilidade de a linguagem atingir precisão absoluta, Gödel, ao contrário, interpreta essa mesma dificuldade como um indício de que a matemática é inesgotável, e que é normal que ela não se deixe circunscrever facilmente.

Os apontamentos feitos em 23 de dezembro de 1929 pelo filósofo Rudolf Carnap, depois de três horas de conversa com Gödel acerca da matemática e do formalismo, confirmam essa visão: "Toda formalização da matemática envolve problemas que se podem exprimir e explicitar na linguagem corrente, mas para os quais não existe expressão apropriada dentro do próprio formalismo. Segue daí (Brouwer foi citado nesse ponto) que a matemática é inesgotável: deve-se sempre retomar a seus inícios, para `buscar nova força nas fontes da intuição'. Não haveria, assim, nenhuma *língua characterístíca uníversalis*, nenhuma língua formal para a totalidade da matemática. (...) Nós dispomos apenas de uma linguagem, mas as sutilezas construídas por nosso espírito são inesgotáveis, porque se baseiam sempre em alguma nova intuição".

O caráter inesgotável da matemática fornecerá a Gödel uma espécie de fio condutor para suas pesquisas ulteriores, especialmente para o teórema da incompletude.

Outro resultado que aparece na tese de Gödel, mas que adquire plena clareza somente no artigo do *Monatshefte,* é o *teorema da compacticidade:* para que um sistema com uma infinidade enumerável de fórmulas seja coerente é necessário e suficiente que cada parte finita do sistema o seja. Esse teorema recebeu pouca atenção quando de sua publicação, sem dúvïda devido aos preconceitos da época em torno da noção de infinito. No entanto, ele iria se tomar, a partir dos anos 40, uma das principais ferramentas conceituais para o desenvolvimento da teoria de modelos.

Gödel evita explicitar em sua tese a noção de verdade. Mais tarde, ele diria que "em razão dos preconceitos filosóficos da época, um conceito de verdade matemática diferente do de demonstrabilidade parecia altamente suspeito e costumava ser rejeitado como desprovido de significado".

Seu teorema da completude representa sem dúvida um resultado notável. Ainda assim, para obter o posto de *privatdozent* que deseja, bem como o acesso à carreira universitária, ele precisa demonstrar alguma "coisa maior". Compreende-se, desse modo, que, apesar de toda a prudência que já havia manifestado em sua tese, o gênio Gödel se dirija precisamente à pedra angular do programa de Hilbert: a demonstração da coerência (não-contradição) da análise matemática. Esse problema é o segundo de uma lista de 23 que Hilbert expôs, em 1900, no Congresso Internacional de Matemática. Tal lista era considerada, à época, como um mapa para toda a matemática do século XX. Aquele que conseguisse

resolver essa questão entraria para o panteão da matemática e teria uma fulgurante carreira universitária.

A idéia de Gödel não era atacar diretamente o problema da não-contradição da análise; mas demonstrar que a análise seria coerente se e somente se a aritmética (teoria dos números) o fosse. Uma vez obtida essa coerência relativa, bastava demonstrar a coerência da teoria dos números, campo em que a utilização de métodos finitos parecia mais promissora. Mesmo assim, o projeto era ousado, pois o método de demonstração da coerência relativa nunca havia sido utilizado fora da geometria (ele havia sido desenvolvido para demonstrar a coerência relativa da geometria não-euclidiana em relação à geometria euclidiana). Gödel penetra, assim, em território desconhecido.

O objetivo de Gödel não é, absolutamente, provocar a queda de todo o programa de Hílbert. Ao contrário, ele se via como um dos matemáticos da nova geração aos quais o grande Hilbert lançara seu apelo apenas dois anos antes, por ocasião do *Congresso de Bolonha.* Aceita, portanto, a tradição de axiomatização da aritmética para elaborar sua demonstração.

Uma axiomatização da teoria dos números havia sido oferecida pelo matemático alemão Richard Dedekind desde 1888 (ver anexo *O método axiomático e as ciências dedutivas*). No entanto, para construir seu sistema de axiomas, Dedekind havia utilizado de maneira informal a teoria dos conjuntos. Mais especificamente, ele colocara no mesmo nível objetos, expressões referidas a objetos e expressões referidas a outras expressões: sua aritmética era de segunda ordem. Deve-se ao matemático italiano Guiseppe Peano a etapa seguinte, decisiva para a axiomatização da matemática. Em sua obra *Arithmetices principia nova methodo exposita,* publicada um ano depois dos trabalhos de Dedekind, Peano apresentou um sistema de axiomas para os números naturais que lembrava de maneira espantosa o sistema de Dedekind, apesar de concebido de modo independente. O matemático italiano, contudo, não construíra sua teoria dentro do contexto conjuntista, e introduziu uma notação (que, com uma ou outra modificação, tornou-se padrão) destinada a contornar certas ambigüidades inerentes à linguagem natural (ver anexo *A Aritmética de Peano*). Seu objetivo era captar, com o maior rigor possível, a natureza lógica do princípio de indução, ou seja, a lógica de segunda ordem.

Em linguagem matemática, o princípio de indução é condensado na fórmula:

$$\forall \alpha (\alpha (0) \wedge \forall \mathbf{x} (\alpha (x) \rightarrow \alpha (\mathbf{s(x)})) \rightarrow \forall \alpha (x)$$

que se lê da seguinte maneira: "para toda propriedade α, se α é válida para zero e se a proposição *se α é válida para um número x, é válida também para seu sucessor´* é verdadeira, então a propriedade α é válida para todo número natural".

Essa frase matemática não é uma fórmula da lógica de predicados de primeira ordem, mas de segunda ordem: com efeito, o primeiro quantificador (o primeiro "todo") não está aplicado a uma variável que representa indivíduos (números naturais), mas a uma variável que representa propriedades desses indivíduos (propriedades dos números naturais). Em seu sistema axiomático, Peano segue exatamente essa formulação do princípio, e especifica que se trata de um axioma de segunda ordem.

4.6.2 Verdade e demonstrabilidade

A distinção entre axiomas de primeira e segunda ordem foi estabelecida pelo lógico polonês Alfred Tarski (1902-1983) para distinguir a linguagem-objeto de um estudo, ou seja, a

linguagem utilizada para falar de um domínio qualquer de objetos, da correspondente metalinguagem, ou seja, a linguagem utilizada para falar da linguagem-objeto. Da mesma maneira, existe uma metametalinguagem, que permite falar da metalinguagem, e assim por diante. Todos esses níveis de linguagem superpõem-se como camadas sucessivas e, para certos estudos lógicos, torna-se essencial separar cuidadosamente cada camada. A façanha de Gödel residirá na invenção de um meio para superar a barreira entre os diferentes níveis de linguagem.

Gödel desejava demonstrar a não-contradição relativa da análise matemática em relação à aritmética de Peano. Essa proposta já o conduz ao âmago do problema verdade-demonstrabilidade: uma proposição verdadeira é sempre demonstrável? No rascunho de uma carta do final dos anos 60, ele descreverá da seguinte maneira seu projeto à época:

"Minha tentativa de demonstração, pela teoria dos modelos, da coerência relativa da análise e da aritmética forneceu também a ocasião para comparar verdade e demonstrabilidade, pois essa demonstração conduz quase obrigatoriamente a tal comparação. Um modelo aritmético para a análise não é nada mais, com efeito, do que uma relação ε que satisfaz ao seguinte axioma de compreensão:

$$\exists \mathbf{N} \ \forall \mathbf{x} (\mathbf{x} \in \mathbf{N} \leftrightarrow \varphi (\mathbf{x}))$$

[Existe um conjunto N tal que, para todo número x, se x é um elemento de N, então a propriedade φ **(x)** é verdadeira, e vice-versa.] Quando se substitui φ **(x)**' por φ **(x)** é demonstrável', tal relação ε toma-se fácil de definir. Dessa forma, se os termos verdade e demonstrabilidade fossem equivalentes, teríamos alcançado nosso objetivo. Segue da correta solução para os paradoxos semânticos, porém, que a verdade das proposições de uma linguagem não poderá jamais expressar-se dentro dessa mesma linguagem, contrariamente à demonstrabilidade (que é uma relação aritmética). Como conseqüência, verdadeiro $\neq$ demonstrável".

Nos anos 30, o problema colocado pela formulação, em uma linguagem dada, de uma definição da noção de verdade para essa mesma linguagem é, certamente, uma das questões mais discutidas nas reuniões do Círculo de Viena*: é possível definir precisamente o significado da expressão "é uma proposição verdadeira na linguagem L"? Em fevereiro de 1930, Menger convida Tarski para uma série de conferências, no curso das quais o matemático polonês sublinha que diversos conceitos utilizados em lógica vêm expressos, não na linguagem-objeto, mas na metalinguagem, e que é importante, portanto, distinguir entre esses dois tipos de linguagem. Nessa ocasião, Gödel solicita a Menger uma conversa particular com Tarski.

Se aceita a argumentação de Tarski acerca do conceito de verdade, Gödel mostra-se mais reticente em relação a esse mesmo tipo de argumentação aplicado à não-contradição e à demonstrabilidade. Tanto assim que abandona a idéia de construir um modelo aritmético para a análise (provavelmente devido às reservas manifestadas na época relativamente à utilização do conceito de verdade em demonstrações) e decide provar que a demonstrabilidade e a não-

* Grupo de pensadores que se reuniam em Viena (1926-1930) de onde surgiu o influente movimento conhecido como positivismo lógico. Organizado por Moritz Schlick (Físico), participavam, entre outros, os filósofos Rudolf Carnap e Karl Popper, o sociólogo e economista Otto Neurath, a matemática Olga Neurath e seu marido, o matemático Hans Hahn, Karl Menger e Kurt Gödel. Entre os estrangeiros que assistiam a algumas reuniões pode-se citar John von Neumann, Willard van Orman Quine, Carl Hemper, Alfred Tarsky e Alfred Jules Ayer.

contradição podem, ainda que indiretamente, ser expressas na linguagem-objeto da teoria, sem que isso acarrete antinomias fatais.

Para o desenvolvimento de seus estudos Gödel concebeu uma interessante formulação de símbolos, fórmulas e provas através de números, bem como mostrou que as proposições metamatemáticas – aliás, sem isso não poderia ter realizado sua prova – podem estar adequadamente refletidas dentro do próprio cálculo, aritmetizando assim a própria metamatemática. No anexo *O Teorema da Incompletude de Gödel* há um pequeno resumo sobre a prova de Gödel.

Gödel acabou com o sonho logicista, visto que não se pode desenvolver toda a aritmética (e muito menos toda a matemática) num sistema que seja ao mesmo tempo consistente e completo. Também acabou com o sonho formalista: existem enunciados matemáticos que são verdadeiros, mas não são suscetíveis de prova, isto é, existe um abismo entre *verdade* e *demonstração**.

Figura 17: Kurt Gödel

4.6.3 Outras conquistas

Gödel também, ao longo da demonstração do seu teorema, rompeu um limiar crucial entre a lógica e a matemática. Ele mostrou que qualquer sistema formal que seja tão rico quanto um sistema numérico qualquer, e que contenha os operadores "+" e "=", pode

* As conclusões de Gödel não significam que seja impossível construir uma prova absoluta e finitista da aritmética. Significam que nenhuma prova deste tipo pode ser construída dentro da aritmética., isto é, que esteja refletida a partir de deduções formais da aritmética. Outras provas metamatemáticas da consistência da aritmética foram construídas, em particular por Gerhard Gentzen, da escola de Hilbert, em 1936, embora não finitistas e não representáveis dentro do cálculo aritmético, ou seja, estão fora das condições previstas por Hilbert.

ser expresso em termos aritméticos **[Coh87].** Isto significa que por mais complexa que se torne a matemática (ou qualquer outro sistema formal redutível a ela), pode-se sempre expressá-la em termos de operações a serem executadas sobre números, e as partes do sistema poderão ser manipuladas por regras de contagem e comparação.

Outro resultado fundamental do teorema da incompletude de Gödel pode-se considerar como sendo a demonstração de que há algumas funções sobre os inteiros que não podem ser representadas por um algoritmo, ou seja, que não podem ser computadas*. Posteriormente verificou-se a existência de uma equivalência entre o Teorema da Incompletude de Gödel e o problema da parada de Turing.

4.7 Alan Mathison Turing: o berço da Computação

Figura 18: Alan Mathison Turing

A revolução do computador começou efetivamente a realizar-se no ano de 1935, em uma tarde de verão na Inglaterra, quando Alan Mathison Turing (1912 - 1954), estudante do King's College, Cambridge, durante curso ministrado pelo matemático Max Neumann, tomou conhecimento do *Entscheidungsproblem* de Hilbert. Enquanto isso, conforme foi brevemente citado no item precedente, uma parte da comunidade dos matemáticos buscava um novo tipo de cálculo lógico, que pudesse, entre outras coisas, colocar em uma base

* Os resultados de Gödel têm conseqüências importantes também para a filosofia. Sabe-se, graças a ele, ser impossível construir uma máquina que, **de modo consistente**, resolva todos os problemas da matemática, com os recursos de um sistema (certos problemas, por assim dizer, "não se deixam resolver" com os recursos do sistema apenas). Mas de fato o matemático os resolve muitas vezes.

matemática segura o conceito heurístico do que seja *proceder a um cômputo*. O resultado destas pesquisas era fundamental para o desenvolvimento da matemática: tratava-se de saber se é possível haver um procedimento efetivo para se solucionar todos os problemas de uma determinada classe que estivesse bem definida. O conjunto desses esforços acabou por formar a fundamentação teórica da que veio a ser chamada "Ciência da Computação".

Os resultados de Gödel e o *problema da decisão* motivaram Turing primeiramente a tentar caracterizar exatamente quais funções são capazes de ser computadas. Em 1936, Turing consagrou-se como um dos maiores matemáticos do seu tempo, quando fez antever aos seus colegas que é possível executar operações computacionais sobre a teoria dos números por meio de uma máquina que tenha embutida as regras de um sistema formal. Turing definiu uma máquina teórica que se tornou um conceito chave dentro da Teoria da Computação. Ele enfatizou desde o início que tais mecanismos podiam ser construídos e sua descoberta acabou abrindo uma nova perspectiva para o esforço de formalizar a matemática, e, ao mesmo tempo, marcou fortemente a História da Computação.

A percepção genial de Turing foi a substituição da noção intuitiva de procedimento efetivo por uma idéia formal, matemática. O resultado foi a construção de uma conceituação matemática da noção de *algoritmo*[*], uma noção que ele modelou baseando-se nos passos que um ser humano dá quando executa um determinado cálculo ou cômputo. Ele formalizou definitivamente o conceito de algoritmo.

4.7.1 A Máquina de Turing

Um dos primeiros modelos de máquina abstrata foi a Máquina de Turing. Conforme o próprio Turing escreveu: "Computar é normalmente escrever símbolos em um papel. Suponha que o papel é quadriculado, podendo ser ignorada a bidimensionalidade, que não é essencial. Suponha que o número de símbolos é finito. [...]. O comportamento do(a) computador(a) é determinado pelos símbolos que ele(a) observa num dado momento, e seu estado mental nesse momento. Suponha que exista um número máximo de símbolos ou quadrículas que ele(a) possa observar a cada momento. Para observar mais serão necessárias operações sucessivas. Admitamos um número finito de estados mentais [...]. Vamos imaginar que as ações feitas pelo(a) computador(a) serão divididas em operações tão elementares que são indivisíveis. Cada ação consiste na mudança do sistema computador(a) e papel. O estado do sistema é dado pelos símbolos no papel, os símbolos observados pelo(a) computador(a) e seu estado mental. A cada operação, não mais de um símbolo é alterado, e apenas os observados são alterados. Além de mudar símbolos, as operações devem mudar o foco da observação, e é razoável que esta mudança deva ser feita para símbolos localizados a uma distância fixa dos anteriores. [...] Algumas destas operações implicam mudanças de estado mental do computador(a) e portanto determinam qual será a próxima ação".

[*] Palavras como procedimento efetivo e algoritmo representam conceitos básicos dentro da Ciência da Computação. São noções que na época de Turing já eram utilizadas pelos matemáticos, como por exemplo Frege e Hilbert (ver capítulos que tratam dessas duas importantes figuras).

O trabalho de Turing ficou documentado no artigo *On Computable Numbers with an aplication to the Entscheidungsproblem*, publicado em 1936* **([Tur50], volume XII)**. Ele descreveu em termos matematicamente precisos como pode ser poderoso um sistema formal *automático*, com regras muito simples de operação.

Turing definiu que os cálculos mentais consistem em operações para transformar números em uma série de *estados* intermediários que progridem de um para outro de acordo com um conjunto fixo de regras, até que uma resposta seja encontrada. Algumas vezes se usa o papel e lápis, para não se perder o estado dos nossos cálculos. As regras da matemática exigem definições mais rígidas que aquelas descritas nas discussões metafísicas sobre os estados da mente humana, e ele concentrou-se na definição desses estados de tal maneira que fossem claros e sem ambigüidades, para que tais definições pudessem ser usadas para comandar as operações da máquina†. Turing começou com uma descrição precisa de um sistema formal, na forma de "tabela de instruções" que especificaria os movimentos a serem feitos para qualquer configuração possível dos estados no sistema. Provou então que os passos de um sistema axiomático formal semelhante à lógica e os estados da máquina que perfaz os "movimentos" em um sistema formal automático são equivalentes entre si. Estes conceitos estão todos subjacentes na tecnologia atual dos computadores digitais, cuja construção tornou-se possível uma década depois da publicação do matemático inglês.

Um sistema formal automático é um dispositivo físico que manipula automaticamente os símbolos de um sistema formal de acordo com as suas regras. A máquina teórica de Turing estabelece tanto um exemplo da sua teoria da computação quanto uma prova de que certos tipos de máquinas computacionais poderiam ser construídas. Efetivamente, uma Máquina de Turing Universal‡, exceto pela velocidade, que depende do hardware, pode emular qualquer computador atual, desde os supercomputadores até os computadores pessoais, com suas complexas estruturas e poderosas capacidades computacionais, desde que não importasse o tempo gasto. Ele provou que para qualquer sistema formal existe uma Máquina de Turing que pode ser programada para imitá-lo. Ou em outras palavras: para qualquer procedimento computacional bem definido, uma Máquina de Turing Universal é capaz de simular um processo mecânico que execute tais procedimentos.

De um ponto de vista teórico, a importância da Máquina de Turing está no fato de que ela representa um objeto matemático formal. Através dela, pela primeira vez, se deu uma boa definição do que significa computar algo. E isso levanta a questão sobre o que exatamente pode ser computado com tal dispositivo matemático, assunto fora do escopo do presente trabalho e que entra no campo da complexidade computacional.

4.7.2 O problema da parada e o problema da decisão

---

* Um ano mais tarde, trabalhando independentemente, Alan Post publicou seu trabalho sobre uma máquina semelhante à de Turing.

† Há um anexo onde se detalha um pouco mais sobre o funcionamento de uma Máquina de Turing.

‡ Uma Máquina de Turing Universal é uma Máquina de Turing específica que lê na sua fita de alimentação, além de dados de entrada, um programa P que é uma especificação de uma Máquina de Turing qualquer.

Turing mostrou que o funcionamento de sua máquina (usar-se-á a sigla MT a partir de agora) e a aplicação das regras de formação de um sistema formal não têm diferença. Ele demonstrou também que seu dispositivo poderia resolver infinitos problemas mas havia alguns que não seriam possíveis, porque não haveria jeito de se prever se o dispositivo pararia ou não. Colocando-se de outra maneira: dado um programa *P* para uma MT e uma determinada entrada de dados *E*, existe algum programa que leia *P* e *E*, e pare após um número finito de passos, gerando uma configuração final na fita que informe se o programa *P* encerra sua execução após um número finito de passos ao processar *E*?

Comparando-se com as afirmações sobre verdades aritméticas, dentro de um sistema formal consistente da aritmética, que não são passíveis de prova dentro deste sistema, percebe-se que o *problema da parada* de Turing nada mais é do que o Teorema de Gödel, mas expresso em termos de uma máquina computacional e programas, ao invés de uma linguagem de um sistema dedutivo da Lógica Matemática.

Em 1936 Turing provou formalmente o seguinte teorema:

Teorema da Parada: Dado um programa ***P*** qualquer para uma Máquina de Turing e uma entrada ***E*** qualquer de dados para esse programa, não existe uma Máquina de Turing específica que pare após um número finito de passos, e que diga se ***P*** em algum momento encerra sua execução ao processar ***E***.

A solução negativa desse problema computacional implica também numa solução negativa para o problema de Hilbert. Portanto nem todos os enunciados verdadeiros da aritmética podem ser provados por um computador*.

4.7.3 Outras participações

4.7.3.1 Decifrando códigos de guerra

Em 1940 o governo inglês começou a interessar-se de maneira especial pelas idéias de Turing. Ele foi convocado pela Escola de Cifras e Códigos, do governo, cuja tarefa era decifrar mensagens codificadas do inimigo. Quando a guerra começou, a Escola Britânica de Códigos era dominada por lingüistas e filólogos. O Ministério do Exterior logo percebeu que os teóricos dos números tinham melhores condições de decifrar os códigos alemães. Para começar, nove dos mais brilhantes teóricos dos números da Inglaterra se reuniram na nova sede da Escola de Códigos em Bletchley Park, uma mansão vitoriana em Bletchley, Buckinghamshire. Turing teve que abandonar suas máquinas hipotéticas com fitas telegráficas infinitas e tempo de processamento interminável, para enfrentar problemas práticos, com recursos finitos e um limite de tempo muito real.

Devido ao segredo que cercava o trabalho realizado por Turing e sua equipe, em Bletchley, a imensa contribuição que prestaram ao esforço de guerra não pôde ser

* Os computadores possuem conjuntos de instruções que correspondem a regras fixas de um sistema formal. Como provou Gödel, existem problemas não solucionáveis dentro de um método axiomático e, portanto, há problemas que um computador não resolve. Esta afirmação não deve ser vista como algo pessimista dentro da Ciência da Computação: que um computador não possa resolver todos os problemas não significa que não se possa construir uma máquina ou algoritmo específico para solucionar **determinado** tipo de problema **[NN56]**.

reconhecida publicamente por muitos anos após o conflito. Costuma-se dizer que a Primeira Guerra Mundial foi a guerra dos químicos e a Segunda Guerra Mundial, a guerra dos físicos. De fato, a partir da informação revelada nas últimas décadas, provavelmente é verdade dizer que a Segunda Guerra Mundial também foi a guerra dos matemáticos. E no caso de uma terceira guerra mundial sua contribuição seria ainda mais crítica.

Em toda sua carreira como decifrador de códigos, Turing nunca perdeu de vista seus objetivos matemáticos. As máquinas hipotéticas tinham sido substituídas por máquinas reais, mas as questões esotéricas permaneciam. Quando a guerra terminou, Turing tinha ajudado a construir um computador, o Colossus, uma máquina inteiramente eletrônica com 1.500 válvulas que eram muito mais rápidas do que os relês eletromecânicos usados nas *bombas*[*]. Colossus era um computador no sentido moderno da palavra. Com sua sofisticação e velocidade extra, ele levou Turing a considerá-lo um cérebro primitivo. Ele tinha memória, podia processar informação, e os estados dentro do computador se assemelhavam aos estados da mente. Turing tinha transformado sua máquina imaginária no primeiro computador legítimo. Depois da guerra, Turing continuou a construir máquinas cada vez mais complexas tais como o Automatic Computing Engine.

Para maiores detalhes sobre os episódios que envolveram Turing e a Máquina Enigma, e de como foi decifrado o código de guerra alemão, ver o anexo *Turing e a Máquina Enigma*.

4.7.3.2 O computador ACE e inteligência artificial

Enviado à América para trocar informações com o serviço de inteligência americano e conhecer os projetos relacionados a computadores, ele tomou conhecimento das emergentes tecnologias eletrônicas e chegou a participar de outro projeto secreto, o *Delilah*, um codificador de voz (conhecido nos filmes de espionagem como *scramblers*), tendo entrado em contato com von Neumann (que quis trazê-lo para junto de si em seus projetos) e com os engenheiros da Bell (incluindo Claude Shannon). De volta para a Inglaterra, entrou para o National Physical Laboratory, onde trabalhou no desenvolvimento do Automatic Computing Engine (ACE), uma das primeiras tentativas de construção de um computador digital. No fim da guerra, já se detinha o conhecimento sobre novas tecnologias eletrônicas, que poderiam ser usadas para aumentar a velocidade dos então circuitos lógicos existentes. A real possibilidade de se construir modelos de Máquinas de Turing Universais fez com que o governo inglês investisse na construção desse novo dispositivo, mas os americanos foram mais agressivos em seus investimentos e acabaram ganhando a corrida na construção de computadores. Turing, vítima de intrigas políticas, ficou fora do centro e controle dos novos trabalhos. Seus relatórios técnicos sobre os projetos de hardware e software do ACE eram ambiciosos e se a

---

[*] A bomba (*Bombe* em inglês) era uma máquina eletromecânica, com vários conjuntos de rotores, idênticos aos da máquina geradora de códigos secretos alemã chamada Enigma(ver o anexo *Turing e a Máquina Enigma*). Ao contrário da Enigma, os rotores da *Bombe* rodavam automaticamente para percorrer todas as configurações possíveis. Quando encontrasse uma configuração que tornasse compatível o palavra adivinhada e o texto cifrado, a máquina parava e o cripto-analista iria testar aquela configuração com o resto do texto cifrado numa Enigma; se o resultado não fosse correcto, re-inicializava a *bombe* para continuar a procura.

máquina originalmente imaginada por ele tivesse sido construída imediatamente, os ingleses não teriam amargado o atraso em relação aos seus colegas do outro lado do Atlântico.

Foi também durante a temporada do ACE que Turing começou a explorar as relações entre o computador e os processos mentais, publicando um artigo, *Computing Machinery and Intelligence* (1950), sobre a possibilidade da construção de máquinas que imitassem o funcionamento do cérebro humano. Pode uma máquina pensar, perguntava-se em seu artigo, e além de focar no assunto inteligência das máquinas, Turing adquiriu especial notoriedade ao tentar introduzir, através desse artigo, um teste para decidir se realmente pode ou não uma máquina pensar imitando o homem. Em novembro de 1991, o Museu do Computador de Boston realizou uma competição entre 8 programas que simulavam o *Turing test*, ganho por um programa chamado PC Therapist III. O problema do teste de Turing é de natureza behaviorista, isto é, somente observa o comportamento exterior, o que lhe dá uma caráter um tanto reducionista. Sérias controvérsias ocorreram e ainda ocorrem sobre esse tema, que esta fora do escopo deste livro*.

4.7.3.3 Programação de computadores

Turing também esteve interessado na programação das operações de um computador – o que então começou a chamar-se de *codificação* – em função das operações matemáticas aí envolvidas e começou a escrever linguagens de programação, avançadas então para o hardware da época. Turing estava convencido de que operações de cálculo eram somente um dos tipos de sistemas formais que poderiam ser imitados pelos computadores. Em particular, ele percebeu como as tabelas de sua máquina teórica poderiam tornar-se elementos de uma poderosa gramática que as máquinas utilizariam para modificar suas próprias operações Turing inovou ao começar a elaborar tabelas de instruções, que *automaticamente* converteriam a escrita decimal a que estamos acostumados em dígitos binários. Estes poderiam ser lidos pelas máquinas que começavam a ser construídas tendo como base a álgebra booleana.

Turing anteviu assim que no futuro, os programadores poderiam trabalhar com as linguagens hoje conhecidas como de *alto nível*. Dizia: "As tabelas de instruções deverão ser feitas por matemáticos com experiência em computadores e certa habilidade em resolver problemas de solução mais difícil. Haverá bastante trabalho deste tipo a ser feito, se todo os processos conhecidos tiverem de ser convertidos na forma de tabelas de instruções em determinado momento. Esta tarefa seguirá paralelamente à construção da máquina, para evitar demoras entre o término desta e a produção de resultados. Poderão ocorrer atrasos, devido a virtuais obstáculos desconhecidos, até o ponto em que será seja melhor deixar os obstáculos lá do que gastar tempo em projetar algo sem problemas (quantas décadas estas coisas levarão?). Este processo de elaboração de tabelas de instruções será fascinante". **[Rhe85]**

Ele percebeu ainda que a capacidade de um computador não estaria somente limitada às questões de hardware, mas também de software. Exceto talvez por Konrad Zuse e

---

* Para uma melhor percepção, existe uma interessante literatura: R.Rucher, Mind Tools; D. Holfstadter, Gödel, Escher, Bach: an eternal golden braid; R. Penrose, The emperor's new mind; J. Lucas, Minds, Machines and Gödel

von Neumann, foi o único a falar sobre os desafios matemáticos e lógicos da arte de programar computadores e seria von Neumann quem completaria em um estilo elegante sua idéia de uma linguagem de programação mais sofisticada.

4.7.4 O triste fim

O trabalho de Turing na Computação e na Matemática foi tragicamente encerrado por seu suicídio em junho de 1954, com a idade de 42 anos. Turing era homossexual e, depois da fuga de dois espiões britânicos de igual tendência para a então União Soviética, nos inícios da década de 1950, houve uma especial pressão sobre ele para corrigir sua condição através do uso de hormônios. Turing, não agüentando a forte pressão, tomou cianeto.

4.8 A tese de Church-Turing e outros resultados teóricos

Até aqui foi mostrado como, do ponto de vista formal, surgiu a idéia de computação. Dentro dessa dimensão formal se procurará mostrar agora que o cume atingido, e ainda não ultrapassado, foi a Máquina de Turing. É um genial modelo abstrato de equipamento, com capacidade de processar complicadas linguagens e calcular o valor de funções aritméticas não-triviais. Pode ainda ser aperfeiçoado para realizar operações mais complexas, embora em relação ao modelo básico isto não implique um salto qualitativo, isto é, que o torne algo mais poderoso.

Em termos computacionais pode-se dizer que as Máquinas de Turing são um modelo exato e formal da noção intuitiva de *algoritmo*: nada pode ser considerado um algoritmo se não puder ser manipulado por uma Máquina de Turing. O princípio de que as Máquinas de Turing são versões formais de algoritmos e de que procedimento computacional algum seja considerado um algoritmo a não ser que possa ser instanciado por uma Máquina de Turing é conhecido como a *Tese de Church*, em homenagem ao brilhante matemático americano Alonzo Church (1903 - 1995), ou ainda *Tese de Church-Turing*. É uma proposição, não um teorema, porque não é um resultado matemático: simplesmente diz que um conceito informal corresponde a um objeto matemático[*].

Fazendo uma pequena retrospectiva. Após os resultados de Gödel, em 1931, muitos lógicos matemáticos partiram em busca do que seria uma noção formalizada de um procedimento efetivo (por efetivo entenda-se *mecânico*), ou seja, o que pode ser feito seguindo-se diretamente um algoritmo ou conjunto de regras (como já visto, antigo sonho de séculos, que remonta a Leibniz). Destas buscas surgiram:

- a sistematização e desenvolvimento das *funções recursivas* (introduzidas nos trabalhos de Gödel e Herbrand) por Stephen Cole Kleene (1909-1994) em sua teoria lógica da

[*] Teoricamente é possível que a tese de Church seja derrubada em algum futuro, caso surja um modelo alternativo de computação que seja publicamente aceitável como algo que preenche totalmente as exigências de executar finitamente cada passo e fazer operações não executadas por qualquer Máquina de Turing. Até a data da confecção deste trabalho não surgiu ainda algo de consistente que viesse a superar a tese de Church (o "computador quântico" – sobre o qual não há ainda uma literatura séria disponível, para se poder falar algo dele nesse trabalho – é algo que poderia ocasionar um abalo nesse sentido)

computabilidade (parte de seu livro *Introdução à Metamatemática*, um dos cumes da lógica matemática dos últimos anos);

- as Máquinas de Turing;
- o cálculo-lambda (componente característico fundamental da linguagem de programação LISP) de Alonzo Church;
- a Máquina de Post, análoga à de Turing, tornada pública um pouco depois, fruto de trabalho independente, e seu sistema para rescrita de símbolos (cuja gramática de Chomsky é um caso particular), de Emil L. Post (1897 - 1954).

Com efeito, todos esses conceitos levaram à mesma conclusão e acabaram por ter o mesmo significado, dentro do citado escopo da busca de uma definição bem elaborada de *processo efetivo*. O que se desenvolverá aqui refere-se mais a Church e Turing (Kleene fez em seu trabalho uma ampla abordagem de ambos, tirando várias conseqüências, e Post trata do mesmo tema de Turing), para se ter uma visão mais clara da diversificação dos estudos da década de 1930, com vistas à fundamentação teórica de toda a Computação.

Em seu célebre teorema, Church demonstrou, em 1936, que não pode existir um procedimento geral de decisão para todas as expressões do Cálculo de Predicados de 1ª ordem, ainda que exista tal procedimento para classes especiais de expressões de tal cálculo. Isso pode causar certo espanto quando se observa que o Cálculo de Predicados de 1ª ordem é semanticamente completo, com o que se diz, implicitamente, que o próprio cálculo, com seus axiomas e regras, constitui um algoritmo capaz de enumerar uma após outra todas as suas expressões válidas. Estas expressões são em quantidade indefinida, e, mesmo sendo enumeráveis (isto é, elaboráveis passo a passo a partir dos axiomas), essa enumeração não tem fim. Compreende-se então que, ao se conseguir demonstrar uma determinada fórmula *P* em certo momento, isto já basta para afirmar que se trata de uma fórmula válida. E, pelo contrário, se depois de haver deduzido mil teoremas dos axiomas, *P* ainda não apareceu, não se pode afirmar nada, porque P poderia aparecer após outros mil teoremas, permitindo-se reconhecer sua validade, ou não aparecer nunca, por não ser válida. Mas não se poderá afirmar em qual caso se está, mesmo depois das mil deduções.**[Aga86]**.

A decisão, dentro desse cálculo, seria possível possuindo-se um algoritmo capaz de enumerar as expressões não válidas. A expressão *P* então aparecia dentro desse conjunto de não válidas em algum momento. O teorema de Church de que se está tratando consiste fundamentalmente na demonstração de que não existe algoritmo capaz de enumerar as expressões não válidas, de maneira que fica excluído a priori todo procedimento de decisão para as expressões do Cálculo de Predicados, em geral. Para compreender as razões de semelhante fato seria necessário valer-se das noções técnicas relacionados com os conceitos da matemática recursiva, que excedem amplamente os limites deste trabalho.

Figura 19: Alonzo Church

Também Church estava interessado no problema de Hilbert. O resultado a que Turing tinha chegado em 1936, sobre o problema da decisão de Hilbert, havia sido também alcançado por Church, alguns poucos meses antes, empregando o conceito formalizado de lambda-definibilidade (ao invés do computável por uma Máquina de Turing, definido por Turing), no lugar do conceito informal *procedimento efetivo ou mecânico*. Kleene, em 1936, mostrou que lambda-definibilidade é equivalente ao conceito de recursividade de Gödel-Herbrand, e, nesse período, Church formulou sua tese, estabelecendo que a recursividade é a própria formalização do efetivamente computável. Isso foi estabelecido, no caso das funções dos inteiros positivos, por Church e Kleene, em 1936.

O cálculo-lambda, como sistema elaborado por Church para ajudar a fundamentar a Matemática (1932/33) era inconsistente, como o mostraram Kleene e Rosser (1935). Mas a parte do cálculo-lambda que tratava de funções recursivas estava correta e teve sucesso. Usando sua teoria, Church propôs uma formalização da noção de "efetivamente computável", através do conceito de lambda-definibilidade. Turing, em 1936 e 1937, ao apresentar a sua noção de computabilidade associada a uma máquina abstrata, mostrou que a noção Turing-computável é equivalente à lambda-definibilidade **[Hur80]**. O trabalho de Church e Turing liga fundamentalmente os computadores com as MT. Os limites das MT, de acordo com a tese de Church-Turing, também descreve os limites de todos os computadores.

O processo que determina o valor de uma função através dos argumentos dessa função é chamado de *cálculo da função* (ou computar uma função). Como foi observado, a máquina de Turing pode ser matematicamente interpretada como um algoritmo e, efetivamente, toda ação de uma máquina algorítmica como o computador pode ser considerada como a de calcular o valor de uma função com determinados argumentos. Este 'insight' é interessante, pois fornece uma maneira de se medir a capacidade computacional de uma máquina. Necessita-se somente identificar as funções que se é capaz de computar e usar

esse conjunto como medida. Uma máquina que compute mais funções que outra é mais poderosa.

A partir dos resultados de Gödel, Turing e Church, pode-se dizer que existem funções para as quais não existe uma seqüência de passos que determinem o seu valor, com base nos seus argumentos. Dizendo-se de outra maneira, não existem algoritmos para a solução de determinadas funções. São as chamadas *funções não computáveis*. Isso significa que para tais funções não há nem haverá capacidade computacional suficiente para resolvê-las. Logo, descobrir as fronteiras entre funções computáveis e não computáveis é equivalente a descobrir os limites do computador em geral. A tese de Church-Turing representa um importante passo nesse sentido.

A percepção de Turing foi a de que as funções computáveis por uma MT eram as mesmas funções computáveis acima referidas. Em outras palavras, ele conjeturou que o poder computacional das MT abarcava qualquer processo algorítmico, ou, analogamente, o conceito da MT propicia um contexto no qual todas as funções computáveis podem ser descritas. Em síntese: as funções computáveis são as mesmas funções Turing-computáveis. A importância disso está na possibilidade de se verificar o alcance e limites de um computador. Na figura que segue pode-se visualizar como se dá a ligação entre os mundos formal, matemático e computacional.

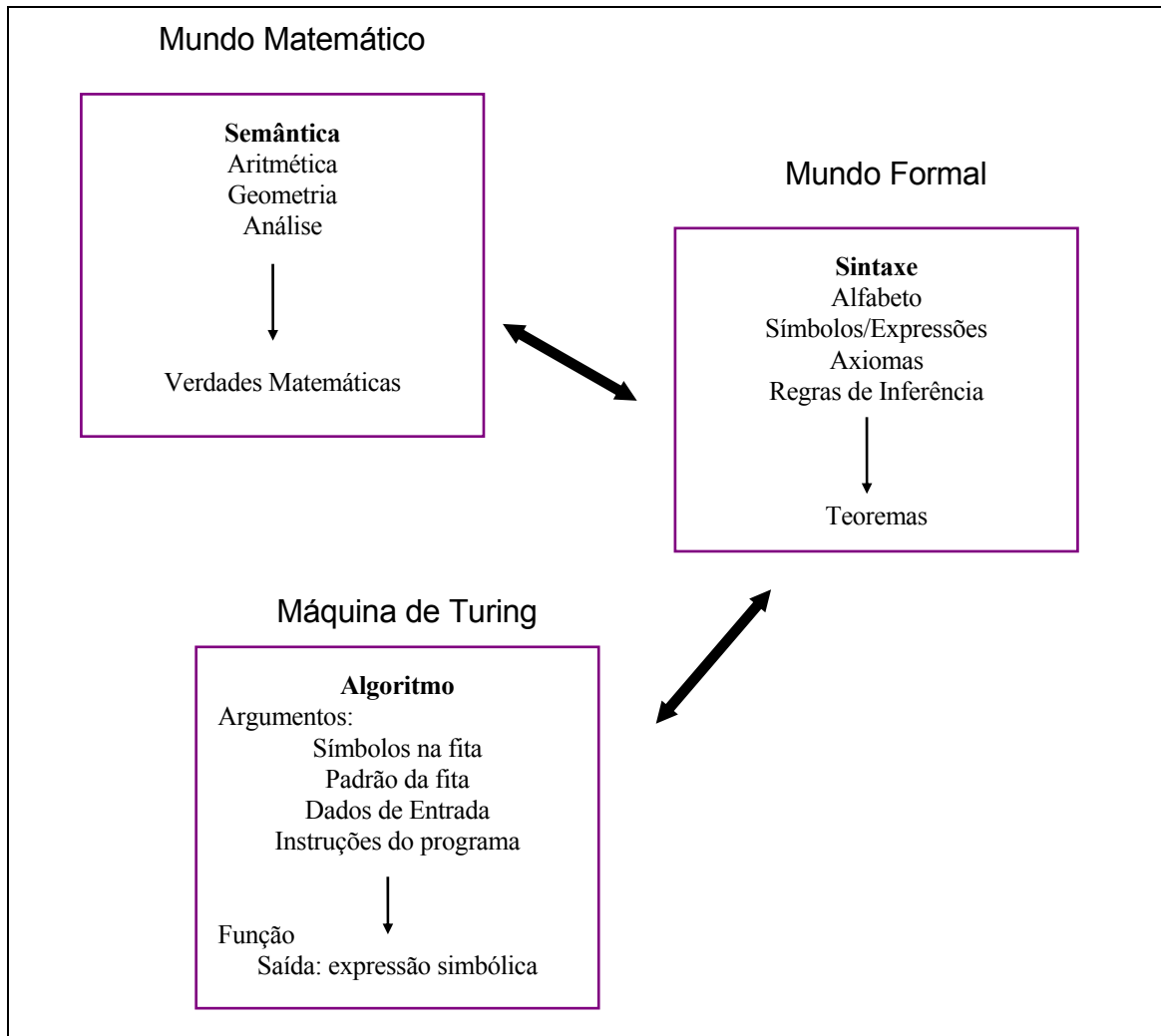



Figura 20: Relacionamento entre mundos formal, matemático e computacional (cfr. **[Cas97]**)

5 Pré-História tecnológica

Como já foi dito, só foi possível chegar aos computadores pelas descobertas teóricas de homens que, ao longo dos séculos, acreditaram na possibilidade de criar ferramentas para aumentar a capacidade intelectual humana, e dispositivos para substituir os aspectos mais mecânicos do modo de pensar do homem. E desde sempre essa preocupação se manifestou na construção de mecanismos para ajudar tanto nos processos de cálculo aritmético quanto nas tarefas repetitivas ou demasiado simples, que pudessem ser substituídas por animais ou máquinas. Neste capítulo se tratará dos dispositivos físicos que precederam o computador, principalmente as máquinas analógicas que incentivaram a corrida final até o aparecimento dos computadores digitais.

5.1 Dispositivos mais antigos

Os primeiros dispositivos que surgiram para ajudar o homem a calcular têm sua origem perdida nos tempos. É o caso, por exemplo, do ábaco e do quadrante. O primeiro, capaz de resolver problemas de adição, subtração, multiplicação e divisão de até 12 inteiros, e que provavelmente já existia na Babilônia por volta do ano 3.000 a.C. Foi muito utilizado pelas civilizações egípcia, grega, chinesa e romana, tendo sido encontrado no Japão, ao término da segunda guerra mundial.

O quadrante era um instrumento para cálculo astronômico, tendo existido por centenas de anos antes de se tornar objeto de vários aperfeiçoamentos. Os antigos babilônios e gregos como, por exemplo, Ptolomeu, usaram vários tipos de dispositivos desse tipo para medir os ângulos entre as estrelas, tendo sido desenvolvidos principalmente a partir do século XVI na Europa. Outro exemplo é o compasso de setor, para cálculos trigonométricos, utilizado para se determinar a altura para o posicionamento da boca de um canhão, e que foi desenvolvido a partir do século XV.

Os antigos gregos chegaram até a desenvolver uma espécie de computador. Em 1901, um velho barco grego foi descoberto na ilha de Antikythera. No seu interior havia um dispositivo (agora chamado de *mecanismo Antikythera*) constituído por engrenagens de metal e ponteiros. Conforme Derek J. de Solla Price, que em 1955 reconstruiu junto com seus colegas essa máquina, o dispositivo Antikythera é "como um grande relógio astronômico sem a peça que regula o movimento, o qual usa aparatos mecânicos para evitar cálculos tediosos" (*An Ancient Greek Computer*, pg. 66[*]). A descoberta desse dispositivo, datado do primeiro século a.C., foi uma total surpresa, provando que algum artesão do mundo grego do mediterrâneo oeste estava pensando em termos de mecanização e matematização do tempo (...)" **[Bol84]**.

5.2 Logaritmos e os primeiros dispositivos mecânicos de cálculo

[*] Trabalho citado por Bolter, que descreve o dispositivo Antikythera, na Scientific American, junho de 1959, pgs. 60-67.

John Napier, Barão de Merchiston, é bastante conhecido pela descoberta dos logaritmos, mas também gastou grande parte de sua vida inventando instrumentos para ajudar no cálculo aritmético, principalmente para o uso de sua primeira tabela de logaritmo.

A partir dos logaritmos de Napier surgiu uma outra grande invenção, desenvolvida pelo brilhante matemático Willian Oughtred e tornada pública em 1630: a régua de cálculo. Ganhou sua forma atual por volta do ano de 1650 (de uma régua que se move entre dois outros blocos fixos), tendo sido esquecida por duzentos anos, para se tornar no século XX o grande símbolo de avanço tecnológico, com uso extremamente difundido, até ser definitivamente substituída pelas calculadoras eletrônicas.

Com o desenvolvimento dos primeiros dispositivos mecânicos para cálculo automático, começa efetivamente a vertente tecnológica que levará à construção dos primeiros computadores. A preparação do caminho para a completa automatização dos processos de cálculo foi executada pelos esforços desses primeiros pioneiros da Computação, que vislumbraram a possibilidade da mecanização mas não possuíam os instrumentos e materiais adequados para concretizar seus projetos. Entre esses grandes nomes não se pode deixar de citar Wilhelm Schickard (1592-1635) e os já citados Pascal e Leibniz. Existem obras sobre essas invenções e somente serão citados os elemento básicos que as compunham*, pois muitas dessas idéias estarão presentes de alguma forma nos futuros computadores.

Quase todas as máquinas para execução de cálculos mecânicos desses três séculos a partir do XVI tinham 6 elementos básicos em sua configuração **[Wil97]**:

- um mecanismo através do qual um número é introduzido na máquina. Nos primeiros projetos isso era parte de um outro mecanismo, chamado seletor, tornando-se algo independente nas máquinas mais avançadas;
- um mecanismo que seleciona e providencia o movimento necessário para executar a adição ou subtração das quantidades apropriadas nos mecanismos de registro;
- um mecanismo (normalmente uma série de discos) que pode ser posicionado para indicar o valor de um número armazenado dentro da máquina (também chamado de registrador);
- um mecanismo para propagar o "vai um" por todos os dígitos do registrador, se necessário, quando um dos dígitos em um registrador de resultado avança do 9 para o 0;
- um mecanismo com a função de controle, para verificar o posicionamento de todas as engrenagens ao fim de cada ciclo de adição;
- um mecanismo de 'limpeza' para preparar o registrador para armazenar o valor zero.

5.3 Charles Babbage e suas máquinas

A idéia de Leibniz de, através de máquinas, liberar o homem das tarefas repetitivas e de simples execução foi quase posta em prática pelo matemático e astrônomo inglês Charles

* Havia também o problema, de modo algum simples, da invenção de mecanismos que produzissem os movimentos exigidos pelas engrenagens durante os cálculos

Babbage (1792-1871), considerado unanimemente um dos grandes pioneiros da era dos computadores. No ano de 1822 ele apresentou em Londres o projeto de um mecanismo feito de madeira e latão, que poderia ter alterado o rumo da história se tivesse sido construído efetivamente. Babbage concebeu a idéia de um dispositivo mecânico capaz de executar uma série de cálculos.

Figura 21: Desenho de Charles Babbage

Já por volta da década de 1820 ele tinha certeza de que a informação poderia ser manipulada por máquina, caso fosse possível antes converter a informação em números. Tal engenho seria movido a vapor, usaria cavilhas, engrenagens, cilindros e outros componentes mecânicos que eram as ferramentas tecnológicas disponíveis em sua época. Para descrever os componentes de sua máquina faltavam-lhe os termos que atualmente são usados. Chamava o processador central de "usina" e referia-se à memória da máquina como "armazém". Babbage imaginava a informação sendo transformada da mesma forma que o algodão – sendo tirada do armazém e modificada para algo diferente. Em 1822 Babbage escrevia uma carta a Sir Humphry Davy, o então presidente da Royal Society, sobre automatizar, como ele próprio dizia, "o intolerável trabalho e a cansativa monotonia" das tabelas de cálculo, escrevendo um trabalho científico intitulado "On the Theoretical Principles of the Machinery for Calculating Tables"(...) **[Gol72]**.

Embora conhecido por seu trabalho na área de Computação, não será demais citar que Charles Babbage foi também um excelente matemático e ao lado de Peacock, Herschel, De Morgan, Gregory e do próprio George Boole, pode ser visto como um dos introdutores da concepção moderna da Álgebra. Além disso foi um dos líderes da Sociedade Real de Astronomia inglesa, tendo publicado também pesquisas no campo da óptica, meteorologia, eletricidade e magnetismo, funcionamento de companhias de apólices de seguros, criptologia, geologia, metalografia, sistemas taxonômicos, máquinas a vapor, etc. Escreveu e publicou

vários livros, um deles (*On the Economy of Machinery and Manufacturers*) reconhecido posteriormente como um dos trabalhos pioneiros na área chamada Pesquisa Operacional.

Mas o que motivou esse inglês a fazer um dispositivo capaz de resolver equações polinomiais através do cálculo de sucessivas diferenças entre conjuntos de números (ver anexo sobre o *Método das Diferenças*) foi a necessidade de uma maior precisão na elaboração de tabelas logarítmicas.

No final do século XVIII houve uma proliferação de tabelas de vários tipos. Desde Leibniz e Newton os matemáticos estiveram preocupados com o problema da produção de tabelas, tanto por meios matemáticos – como no caso das de multiplicação, seno, coseno, logaritmos, etc. – ou por meio de medições físicas – densidade em função da altitude, constante gravitacional em diferentes pontos da terra, entre outras coisas. A intenção era reduzir o trabalho de cálculo, mas as tabelas produzidas pelos especialistas tinham muitos erros. Os matemáticos estavam cientes deles e estudos foram elaborados para se tentar melhorar a situação. Nestas circunstâncias apareceu o projeto denominado *Difference Engine* de Babbage, que lhe valeu o apoio de seus colegas da Sociedade Real e fundos do governo britânico para iniciá-lo.

O desafio era construir um dispositivo para computar e imprimir um conjunto de tabelas matemáticas. Babbage contratou um especialista em máquinas, montou uma oficina e então começou a descobrir quão distante estava a tecnologia do seu tempo daqueles mecanismos altamente precisos e de movimentos complexos exigidos pelo seu projeto. A conclusão foi que deveria, antes de iniciar a construção da Máquina de Diferenças, gastar parte dos seus recursos para tentar avançar o próprio estado da arte da tecnologia vigente. Todos estes trabalhos prolongaram-se por alguns anos, sem sucesso, até que o governo inglês desistiu do financiamento. Em 1833 Charles Babbage parou de trabalhar em sua máquina*.

Apesar de tudo, esse teimoso inglês já vinha desenvolvendo novas idéias. Provavelmente tentando alguma nova modificação no projeto da Máquina de Diferenças foi que Charles Babbage concebeu um mecanismo mais complicado que este em que falhara após vários anos de tentativas. O pensamento era simples: se é possível construir uma máquina para executar um determinado tipo de cálculo, por que não será possível construir outra capaz de *qualquer* tipo de cálculo? Ao invés de pequenas máquinas para executar diferentes tipos de cálculos, não será possível fazer uma máquina cujas peças possam executar diferentes operações em diferentes tempos, bastando para isso trocar a ordem em que as peças interagem?

Era a idéia de uma máquina de cálculo universal, que virá a ser retomada em 1930 por Alan Turing, e que terá então conseqüências decisivas. Vale ressaltar que o *Analitical Engine*, a Máquina Analítica – nome dado por Charles Babbage à sua invenção – estava muito próxima conceitualmente daquilo que hoje é chamado de computador.

A Máquina Analítica poderia seguir conjuntos mutáveis de instruções e, portanto, servir a diferentes funções – mais tarde isso será chamado de *software*... Ele percebeu que para

---

* Esta máquina, conforme imaginada por Babbage, foi construída e colocada em operação pelo Museu de Ciência de Londres e mostrada com seus desenhos em 1862 durante exposição internacional. Em 1849 Babbage entregaria ao governo britânico uma nova versão da Máquina de Diferenças, que nem considerada foi. Em 1991 foi construida esta segunda versão **[Wil97]**.

criar estas instruções precisaria de um tipo inteiramente novo de linguagem e a imaginou como números, flechas e outros símbolos. Ela serviria para Babbage "programar" a Máquina Analítica, com uma longa série de instruções condicionais, que lhe permitiriam modificar suas ações em resposta a diferentes situações.

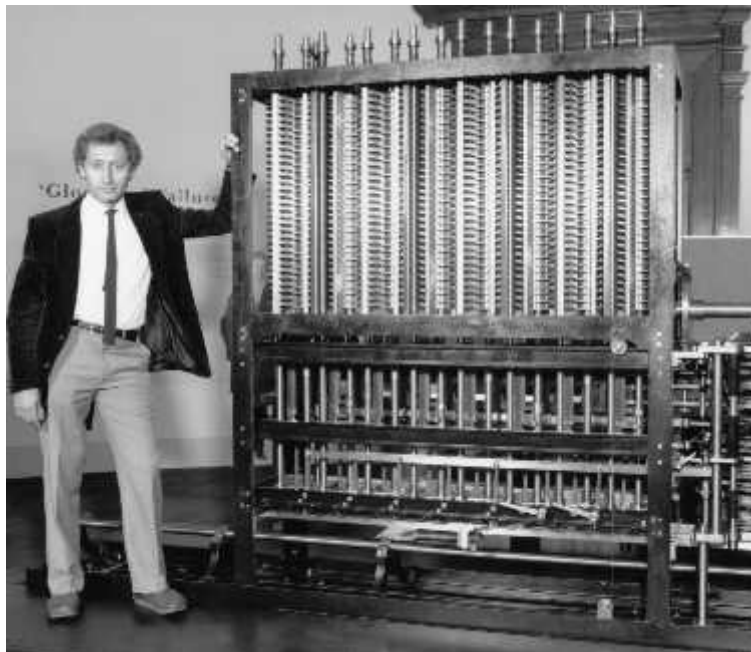



Figura 22: Máquina Diferencial de Babbage construída pelo Museu de Londres

Reconhecendo a importância de se terem resultados impressos, Charles procurou que os resultados finais e os intermediários fossem impressos para evitar erros. Dispositivos de entrada e saída eram assim necessários. A entrada de dados para a máquina seria feita através de três tipos de cartões: "cartões de números", com os números das constantes de um problema; "cartões diretivos" para o controle do movimento dos números na máquina; e "cartões de operação" para dirigir a execução das operações tais como adições, subtrações, etc. Mas o mais genial estava por vir: duas inovações simples mas que produziram um grande impacto. A primeira era o conceito de "transferência de controle" que permitia à máquina comparar quantidades e, dependendo dos resultados da comparação, desviar para outra instrução ou seqüência de instruções. A segunda característica era possibilitar que os resultados dos cálculos pudessem alterar outros números e instruções colocadas na máquina, permitindo que o "computador" modificasse seu próprio programa. Nestes temas teve importante participação, Ada Augusta Byron, condessa de Lovelace, a primeira efetiva programadora de computadores, sobre a qual ainda se falará.

5.3.1 A máquina de Jacquard, inspiração de Babbage

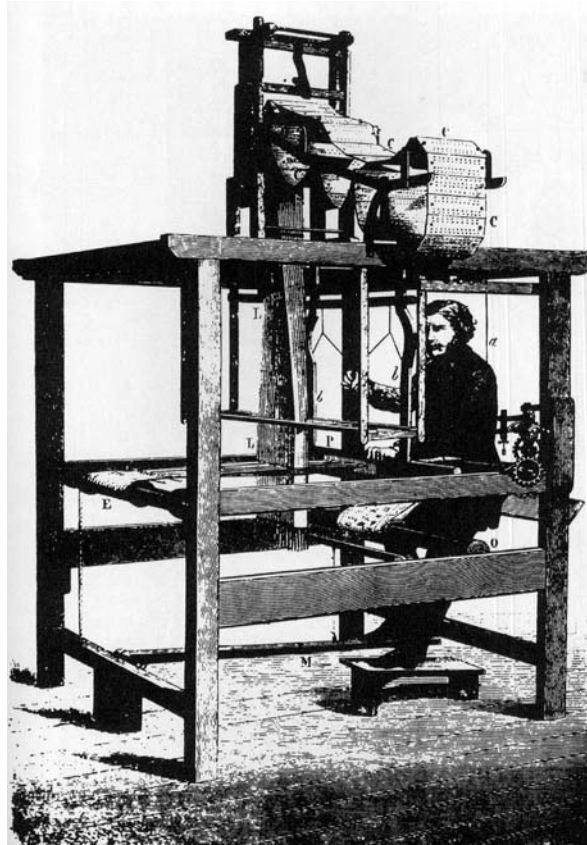



Figura 23: Tear de Jacquard

É importante fazer uma menção a Joseph-Mariae Jacquard (1752-1834), um francês que produziu uma máquina para substituir o trabalho humano. Na verdade, Babbage despertou para seu novo projeto observando a revolução produzida pelos teares de Jacquard[*], dotados de um dispositivo que automatizava o processo de tecelagem com vistas a obter determinados padrões de desenho. Para executar um determinado trançado, a fiandeira deveria ter um plano ou *programa* que lhe dissesse que fios deveriam passar por cima ou por baixo, quando repetir o processo, etc. O ponto chave da máquina de Jacquard era o uso de uma série de cartões cujos buracos estavam configurados para descrever o modelo a ser produzido. O sucesso foi total e em 1812 havia na França cerca de 11.000 teares de Jacquard **([Bri79b], volume V)**. Adaptando o tear de Jacquard, a Máquina Analítica processava padrões algébricos da mesma maneira que o tear processava padrões de desenhos.

5.3.2 Uma Lady como primeira programadora

Ada Augusta Byron era filha do famoso poeta Lord Byron e foi educada pelo matemático logicista inglês Augustus De Morgan. Bem cedo demonstrou ter grandes talentos

[*] O tear de Jacquard inspirou também a Herman Hollerith, sobre quem se falará mais adiante.

na área. Apresentada a Babbage durante a primeira demonstração da Máquina de Diferenças, tornou-se uma importante auxiliar em seu trabalho, sendo, sobretudo, alguém que compreendeu o alcance das novas invenções. Ela percebeu que, diferentemente das máquinas anteriores com funcionamento analógico (execução de cálculos usando *medidas*), a Máquina de Diferenças era *digital* (execução de cálculos usando *fórmulas* numéricas). Mais importante ainda, deu-se conta da combinação entre funções lógicas e aritméticas na máquina de Babbage.

Quando Charles Babbage visitou Turim a convite do amigo Giovanni Plana, astrônomo e compilador de tabelas, ministrou uma série de palestras para distintos públicos, incluindo Luigi F. Menabrea, futuro primeiro-ministro da Itália. Este ficou impressionado com o trabalho de Babbage e tomou uma série de notas, publicadas depois em 1842 pela Biblioteca da Universidade de Genebra. Lady Lovelace traduziu para o inglês essas notas, acrescentando muitas observações pessoais **[Gol72]**. Esta publicação e outro ensaio (*Observations on Mr. Babbage's Analytical Engine*) a colocam como patrona da arte e ciência da programação. Conforme comentado por B.H. Newman, os escritos de Ada Byron "mostram como ela teve uma total compreensão dos princípios de um computador programado, um século antes do tempo deste" **[Moo77]**.

Mesmo não estando a máquina construída, Ada procurou escrever seqüências de instruções tendo descoberto conceitos que seriam largamente utilizados na programação de computadores como *subrotinas, loops e saltos*.

5.4 Outras Máquinas Diferenciais e Máquinas Analíticas

Embora não fosse fácil, o trabalho de Babbage foi divulgado por um certo Dr. Dionysus Lardner, que procurou descrever a máquina e seu modo geral de operação **[Wil97]**. Um sueco, George Scheutz, editor de um jornal técnico de Estocolmo, leu e ficou entusiasmado pela máquina descrita por Lardner e, sem se comunicar com Babbage, propôs-se a construir a sua Máquina de Diferenças, junto com o filho[*]. Os anos de 1840, 1842 e 1843 marcaram etapas bem sucedidas no desenvolvimento do projeto, culminando com um modelo preliminar. Em outubro de 1854 o dispositivo de Scheutz estava completo e em funcionamento.

Outros, como por exemplo Alfred Decon, inglês, Martin Wiberg, sueco e G. B. Grant, americano, construíram modelos derivados, e até 1931 Máquinas de Diferenças foram construídas para produzir diferentes tipos de tabelas **[Wil97]**.

Com relação à Máquina Analítica, parece que o irlandês Percy Ludgate (1883-1922) projetou e tentou construir um mecanismo similar ao de Babbage, conforme pequena descrição feita em um diário científico de Dublin, em 1909.

[*] Em 1854, durante uma viagem a Londres, Scheutz pai e filho encontraram-se com Charles Babbage, que aprovou a máquina por eles construída. Ambos nunca esconderam depois sua admiração pelas idéias do inglês.

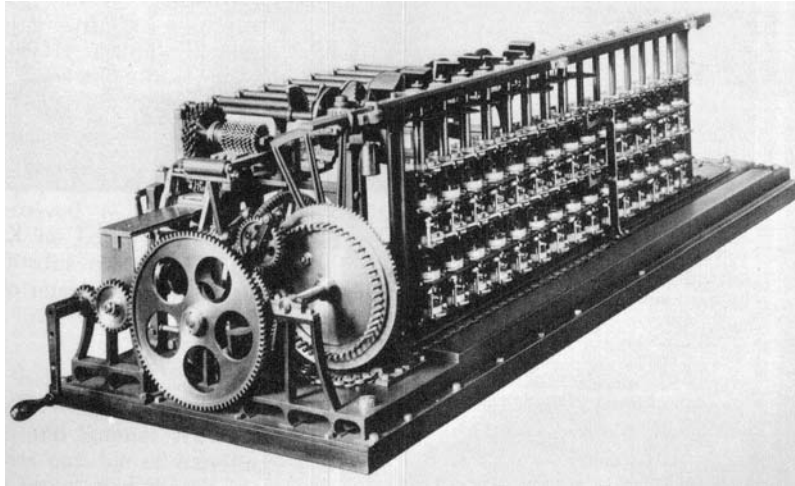



Figura 24: Máquina Diferencial de George Scheutz

5.5 A última contribuição do século XIX: Herman Hollerith

Figura 25: Tabuladora de Hollerith

O próximo passo importante na História da Computação não está relacionado com tabelas de cálculo de logaritmos ou desenvolvimento de leis do pensamento. O próximo "pensador" a avançar o estado da arte foi Herman Hollerith, um empregado de apenas 19 anos do United States Census Office. Seu papel não teve impacto sobre os importantes fundamentos teóricos da Computação e sua invenção já é obsoleta. Mas sua pequena inovação cresceu tanto na indústria que, mais tarde, Hollerith veio a dominar o uso da

tecnologia de computadores. Em 1890 ele ganhou a concorrência para o desenvolvimento de um equipamento de processamento de dados para auxiliar o censo americano daquele ano. A empresa fundada para isto, Hollerith Tabulating Machines, veio a ser uma das três que em 1914 compôs a empresa CTR (Calculating-Tabulating-Recording), renomeada em 1924 para International Business Machine - IBM **[IEEE95]**.

Hollerith, inspirado pelos teares de Jacquard, desenvolveu a idéia de se aproveitar os cartões perfurados dos teares em uma máquina que pudesse interpretar, classificar e manipular as somas aritméticas representadas pelas perfurações. Ele combinou cartões perfurados com os novos dispositivos eletromagnéticos de então.

5.6 Computadores analógicos

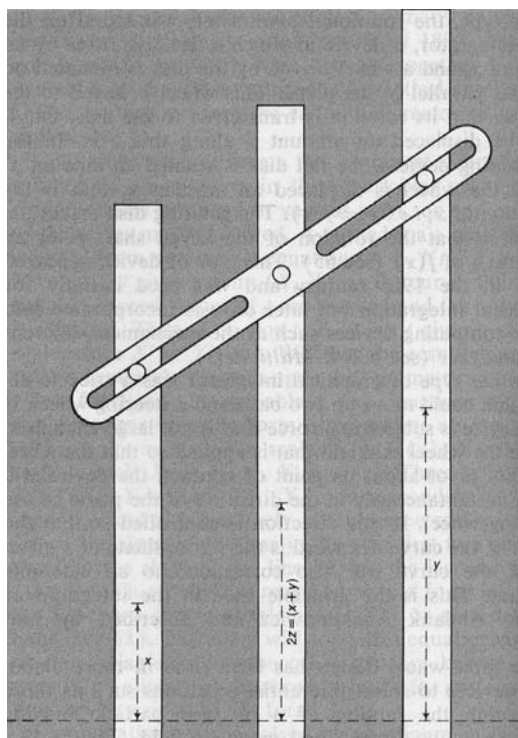



Figura 26: Dispositivo analógico simples

Há uma história interessante sobre os computadores analógicos, cujas origens remontam ao passado distante. Muitos dispositivos analógicos foram desenvolvidos a partir do ano 400 a.C. Típicos são os astrolábios (ver anexo sobre o assunto), o já mencionado mecanismo Antikythera, os instrumentos de sinalização e os planetários **[Wil97]**. Irá interessar particularmente para esse trabalho uma classe específica de instrumentos analógicos: as máquinas integradoras, que remontam a Maxwell, Faraday, Kelvin e Michelson, entre outros, que tentaram desenvolver dispositivos para executar operações matemáticas **[Lil45]**. Essas foram usadas em projetos que exigiam a solução de equações diferenciais e modelagem de sistemas mais complexos, como o movimento das ondas do mar, evoluindo até os computadores eletrônicos analógicos, alguns ainda usados até os dias de hoje para aplicações especiais. Tais desenvolvimentos formam uma parte dessa infra-estrutura que constituiu a base para o aparecimento dos computadores digitais.

Um computador analógico é um dispositivo no qual os números são representados por quantidades físicas medidas, e nos quais equações ou relações matemáticas são representadas por diferentes componentes, correspondendo a operações matemáticas singulares, tais como integração, adição ou multiplicação.

Um dispositivo analógico muito conhecido é a régua de cálculo. Ela consiste basicamente de dois trilhos graduados de acordo com os logaritmos de números, e os trilhos deslizam um sobre o outro. Os números são representados através de comprimentos nos trilhos e a operação física que pode ser executada é a soma de dois comprimentos nos trilhos. Sabe-se que o logaritmo de um produto de dois números é a soma dos logaritmos deles. Assim pode-se com a régua de cálculo formar a soma de dois comprimentos e executar multiplicação e operações correlatas.

Os componentes analógicos podem ser divididos em duas classes, dependendo da maneira como os números são representados: i) por quantidades mecânicas, como um deslocamento linear ou rotação angular; ii) quantidades elétricas, como voltagem, corrente, impedância, condutividade.

Se os deslocamentos lineares são usados para representar números, há caminhos simples, nos quais relações geométricas podem aparecer através de formas mecânicas. As operações matemáticas podem ser realizadas usando-se uma relação geométrica correspondente. Na figura ao lado pode-se ver um computador analógico muito simples.

No final do século XIX, as equações matemáticas que apareciam nos estudos de física passaram a exigir uma grande quantidade de cálculos, quase impossíveis de se resolver na prática. Os físicos começaram a desenvolver sofisticadas ferramentas matemáticas para descrever, através de equações*, a operação de determinados tipos de mecanismos, assim como conceber máquinas cujo movimento era feito de acordo com equações. Uma solução foi a de se criar um sistema físico análogo e cujo comportamento pudesse ser quantitativamente observado. Por exemplo: o fluxo de calor é análogo ao fluxo de eletricidade, onde temperatura corresponde a potencial elétrico. Logo, pela análise de camadas eletricamente condutoras, dispostas de maneira a simular às características de uma estrutura, pode-se investigar o fluxo de calor dentro dessa estrutura **([Bri79a], volume XI)**. Alguém que quisesse projetar um dispositivo desse tipo deveria:

i) analisar quais operações desejaria executar;
ii) procurar um aparato físico cujas leis de operação sejam *análogas* àquelas que se deseja executar;
iii) construir o aparelho;
iv) resolver o problema *medindo* as quantidades físicas envolvidas.

Dois nomes famosos estão diretamente ligados à efetiva produção de dispositivos analógicos para resolução de cálculos mais complexos: James Clerk Maxwell (1831-1879), o criador da teoria sobre a eletricidade e o magnetismo, e James Thomson. Ambos inventaram dispositivos analógicos por volta de 1860 **[Gol72]**.

---

* Como por exemplo equações diferenciais ordinárias, séries de transformações de Fourier, sistemas de equações algébricas lineares

Em todos os dispositivos analógicos que começaram a aparecer, a operação fundamental é a da integral, isto é, todos eles produziam como saída $\int_a^b f(x)dx$, dado f(x) como entrada.

Dentro da evolução das máquinas analógicas, os analisadores diferenciais foram os dispositivos que mais tarde passaram a ser chamados propriamente de computadores analógicos.

5.6.1 Primeiras evoluções: século XV

É por volta do século XV que aparecem dispositivos analógicos mais sofisticados, utilizados para prever os intervalos de tempos de maré alta e baixa em alguns portos europeus. São os chamados "tide predictors", com suas escalas circulares, seus ponteiros que marcavam a posição do sol e da lua – e um interessante sistema de checagem desses dados – e que, juntamente com algumas informações específicas do porto, permitia ao usuário ler nas escalas do instrumento o tempo aproximado entre a maré alta e baixa. Quando na metade do século XVIII foi possível encontrar uma fórmula para o cálculo de séries de coeficientes de coseno (y = A cos(u) + B cos(v) + C cos(w) + ...), Lord Kelvin construiu uma máquina analógica para avaliar essa fórmula. Chamou-a *analisador harmônico*, e um exemplo pode ser visto na próxima figura.

Um desses primeiros dispositivos foi elaborado em 1878. Escrevendo sobre seu analisador harmônico de ondas do mar Kelvin disse: "O objetivo desta máquina é substituir o grande trabalho mecânico de calcular os fatores elementares que constituem a subida e descida da maré (...)"**[Gol72]**. Uma análise harmônica consiste em se formar um número de integrais do tipo geral $\int f(t)g(t)dt$, onde *g* é uma função seno ou coseno. A avaliação das integrais desse tipo foi o que Kelvin conseguiu, fazendo uma engenhosa adaptação de um integrador* elaborado por seu irmão.

A última invenção de Kelvin relevante para nossa história foi o que agora é chamado *Analisador Diferencial*, um dispositivo para a solução de sistemas de equações diferenciais ordinárias. Dos dispositivos chamados integradores é possível obter uma integral que é o produto de duas variáveis. Uma grande gama de sistemas de equações pode ser computada por esses componentes. Kelvin nunca chegou a construir sua máquina por não dispor de tecnologia suficiente. A dificuldade estava em como usar a saída de um integrador como entrada em outro. Na explicação de Maxwell, o problema central era a saída estar medida pela rotação de um disco ligado a uma roda. Esta roda é acionada por estar apoiada sobre um disco que gira em torno de um eixo. O torque desse disco − sua capacidade de girar a roda − é muito pequeno e conseqüentemente ele, de fato, não pode fornecer uma entrada

* Integrador é também um dispositivo analógico, que produz como resultado a integral de f(x). Seria exaustivo e fugiria do escopo do trabalho falar sobre esses dispositivos – existem ainda os planímetros, para medir áreas de figuras traçadas por um operador humano, etc. – que fazem parte desses primeiros esforços em direção a sofisticados mecanismos analógicos.

para outro integrador*. Esses problemas permaneceram suspensos por quase 50 anos até o desenvolvimento dos amplificadores de torque. Analisadores diferenciais mecânicos foram revitalizados por volta de 1925 e o mais famoso destes foi o construído no Instituto de Tecnologia de Massachusetts (MIT) por Vannevar Bush†.

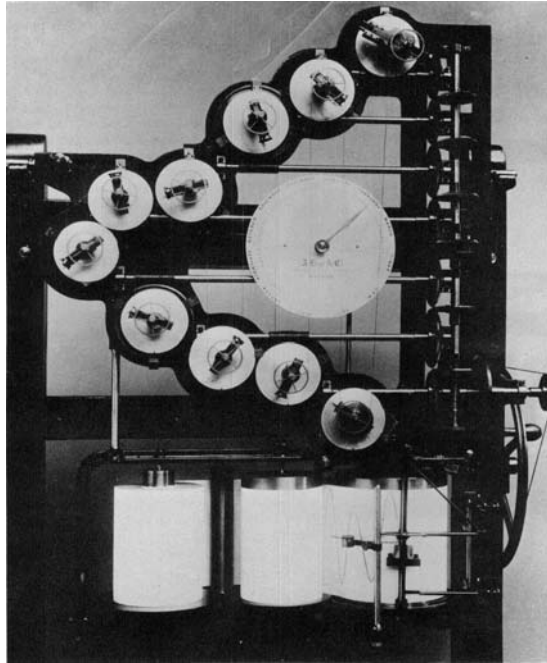



Figura 27: Dispositivo analógico de Lord Kelvin

5.6.2 Michelson e seu analisador harmônico; I Guerra Mundial

"O principal obstáculo na construção de tal máquina está na acumulação de erros envolvida no processo de adição. O único instrumento projetado para efetuar esta adição é o de Lord Kelvin (...). O alcance da máquina é, no entanto, limitado pelo pequeno número de elementos na conta (...), pois com um considerável aumento no número de elementos, os erros acumulados devido aos fatores já mencionados logo neutralizariam as vantagens do aumento do número de termos na série."

São palavras de Albert A. Michelson (1852 - 1931) em 1898 **[MS98]**, um dos grandes físicos do século XX. Interessou-se pelo desenvolvimento de um analisador harmônico que pudesse manipular uma série de Fourier de até 20 termos, continuando a tradição das máquinas analógicas.

* Outra dificuldade substancial: não é possível aumentar muito o número de termos em uma série pois o seu dispositivo de adição de termos levava a um acúmulo de erros. Para uma longa série de termos o resultado poderia estar completamente viciado.

† Após a Segunda Guerra Mundial, analisadores diferenciais mecânicos começaram a se tornar obsoletos com o desenvolvimento de analisadores diferenciais eletrônicos e com o aparecimento da Computação eletrônica digital.

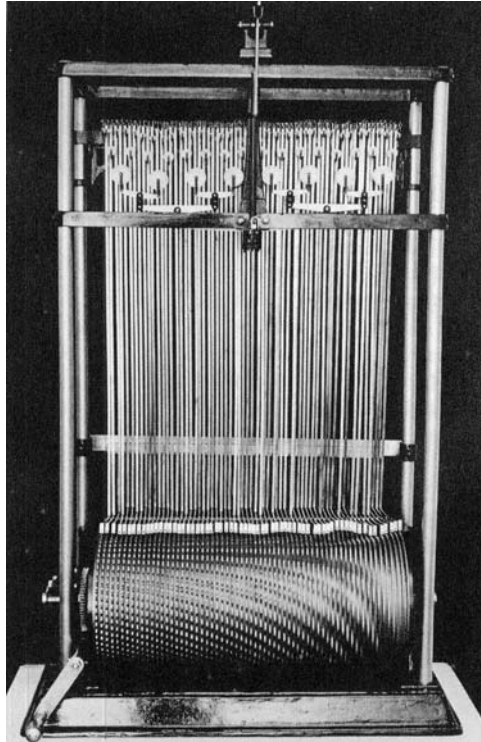



Figura 28: Analisador harmônico de Michelson

Durante a I Guerra Mundial tornaram-se estratégicos os problemas referentes aos cálculos balísticos, o que foi um incentivo à continuidade do desenvolvimento de máquinas computacionais. Um desses problemas é o de como determinar a função de deslocamento, observando-se a resistência do ar, em função da velocidade. Quando a artilharia aponta para objetos que se movem, como navios ou aviões, é essencial prever o movimento dos alvos.

Foram duas décadas (1910 e 1920) em que houve um grande aprofundamento teórico, com a formação de grupos de matemáticos nos EUA e Inglaterra, cujas principais descobertas estão nos procedimentos numéricos para solução de equações diferenciais com grande precisão **[Gol72]**.

5.6.3 Computadores analógicos eletromecânicos

Nos primeiros anos do século XX muitos físicos e engenheiros de todo o mundo estiveram trabalhando em questões fundamentais da área de eletricidade. Centros de pesquisa foram criados em Harvard, no MIT, na IBM, na General Electric, e outros lugares. Tiveram sucesso na formulação matemática dos problemas em teoria de circuitos e muitos textos foram escritos nos anos da década de 1920, especialmente por Vannevar Bush no MIT, A.E. Kennelly de Harvard e do MIT, C.P Steinmetz da General Electric, entre outros **[Gol72]**. Também não se pode esquecer o trabalho fundamental de Oliver Heaviside (1850-1897), um inglês que desenvolveu um dispositivo matemático para manipular equações e analisar indução eletromagnética, e o trabalho de Norbert Wiener junto a Bush.

Como se disse sobre Kelvin e seu analisador harmônico, o grande problema foi ele não dispor da suficiente tecnologia para desenvolver um dispositivo que executasse a

operação de gerar a integral do produto de duas funções, $\int_a^b f1(x)f2(x)dx$, e por vários anos a idéia esteve esquecida até o desenvolvimento dos amplificadores de torque.

A partir de 1927 até 1931, Vannevar Bush e sua equipe no MIT desenvolveram mecanismos para resolver equações diferenciais ordinárias. Bush deve especialmente a C. W. Niemann, engenheiro e inventor do amplificador de torque Bethlehem, a possibilidade de ter construído seu famoso analisador diferencial, terminado em 1931. Usando o amplificador de Niemann, Bush pôde construir uma máquina usando exclusivamente integradores. Ainda mecânico, este dispositivo foi aprimorado durante a II Guerra Mundial, pela substituição dos mecanismos puramente mecânicos por corrente e voltagem, obtidas através de potenciômetros instalados sobre os discos cuja rotação representava quantidades. As voltagens correspondiam à soma, produto e a uma função de uma variável. Entram aqui conceitos de servo-mecanismos e amplificadores operacionais **[Ryd67]**.

Ainda dentro do mundo dos computadores analógicos, deve-se destacar o trabalho do físico inglês Douglas Hartree, das universidades de Manchester e Cambridge, que tentou resolver equações diferenciais parciais com analisadores diferenciais, e que, ao deparar-se com cálculos altamente complexos, anteviu e preparou o advento dos computadores eletrônicos **[Gol72]**.

As novas descobertas da indústria e da ciência no campo da eletricidade – proporcionando rapidez e precisão aos equipamentos – juntamente com a limitação dos equivalentes analógicos eletromecânicos, acabaria por impor a nova tecnologia de circuitos. Uma nova era da Computação começava a ser desvelada. É necessário assinalar, no entanto, que novas máquinas analógicas eletromecânicas – sucedâneas da última máquina de Bush, no MIT, em 1942 – foram construídas e até 1960 ainda estavam em uso **([Bri79a], volume XI)**.

5.7 Circuitos elétricos e formalismo lógico: Claude Elwood Shannon

Como um grande tapete, que vai sendo tecido aos poucos por diferentes artesãos que não têm a visão de todo o conjunto, paulatinamente avançou a teoria e a técnica que levaram à construção do computador digital. Paralelamente aos matemáticos, também um jovem engenheiro, Claude E. Shannon (1916 - 2001), com a idade de 22 anos, deu uma grande contribuição à Computação: em 1937 ele estabeleceu uma ligação entre os circuitos elétricos e o formalismo lógico. Mestre em Engenharia Elétrica e Doutor (PhD) em Matemática pelo Instituto de Tecnologia de Massachusetts (MIT), Cambridge, MA, em 1940, durante a Segunda Guerra Mundial, Shannon começou a desenvolver uma descrição matemática da informação, dando origem a um ramo de estudos conhecido como *Teoria da Informação*[**Gat95**]. Deu ainda importantes contribuições na área da Inteligência Artificial.

Figura 29: Claude E. Shannon

O que Shannon fez em 1937 foi mostrar um caminho para projetar máquinas baseadas na lógica algébrica descrita um século antes por George Boole, aquela em que só havia dois valores no sistema de cálculo lógico: 1 e 0. Se um valor é verdadeiro, ele pode ser representado pelo valor 1 e, se falso, pelo 0. Nesse sistema, uma *tabela-verdade* descreveria os vários estados lógicos possíveis. Uma das características importantes da álgebra de Boole é que as operações lógicas podem ser colocadas juntas e formar novas operações.

Claude Shannon percebeu que a mesma álgebra poderia descrever o comportamento de circuitos elétricos chaveados. Igualmente importante foi o modo como estas combinações entre operações lógicas e aritméticas poderiam ser usadas para se construir uma "operação de memória". A álgebra booleana torna possível a construção de um dispositivo de "estado" que pode armazenar qualquer informação específica, seja um dado ou uma operação. E se um circuito elétrico pode executar operações matemáticas e lógicas, e pode também armazenar os resultados de tais operações, então os computadores digitais podem ser construídos.

Em resumo:

- lógica booleana, cujas tabelas-verdade poderiam representar as regras de um sistema lógico formal;
- tabelas de instruções da Máquina de Turing que podem simular as tabelas-verdade de Boole;
- dispositivos como o relê – já então muito usados em telefones – para representar "estados" de máquina.

Em breve já seria possível a construção de circuitos elétricos que simulavam algumas operações lógicas. Shannon estava procurando um procedimento matemático que fosse o mais adequado para se descrever o comportamento de circuitos a relê*. Sua tese de mestrado publicada em 1937 mostrou como a álgebra booleana poderia ser usada para descrever as operações desses complexos circuitos.

* Um relê é uma chave ou dispositivo que abre ou fecha um circuito, permitindo ou bloqueando o fluxo da eletricidade. É semelhante a um interruptor de luz, com a diferença de que o relê não é ligado ou desligado por uma ação humana, mas pela passagem de uma corrente elétrica.

Nos dez anos seguintes ao seu primeiro trabalho (a tese anteriormente citada), Shannon dirigiu seu interesse para o estudo da comunicação, parte de um trabalho já iniciado por Norbert Wiener, de quem se falará mais adiante.

Depois da guerra, tendo encontrado uma ferramenta perfeita para a descrição de circuitos a relê, Claude Shannon procurou definir matematicamente aquilo que as novas máquinas processavam. Ele estava interessado nas leis subjacentes aos sistemas de comunicação de mensagens feitos pelo homem, na diferença entre ruído e mensagem e de como esta mantinha a sua ordem em um meio onde a "desordem" – ruído – é muito alta. Chegou a equações muito parecidas às do físico *Boltzmann* sobre as leis da entropia.

Em 1948 Shannon publicou dois trabalhos que originaram a já citada área da Teoria da Informação (A Mathematical Theory of Information)[*]. O desenvolvimento deu-se rapidamente, afetando não somente o projeto de sistemas de comunicação, mas também áreas como automatização, ciência da informação, psicologia, lingüística e termodinâmica **([Bri79a], volume IX)**.

Em 1950 publicou "A Chess Playing Machine" onde propunha que computadores digitais poderiam ser adaptados para trabalhar simbolicamente com elementos representando palavras, proposições ou outras entidades conceituais, dando prosseguimento ao emergente ramo de estudos denominado mais tarde Inteligência Artificial. Em 1953, com "Computers and Automata" falou sobre simulação, através de hardware e software, de algumas operações da mente **([Rhe85], capítulo 6)**.

Em 1956, mantendo seu trabalho nos laboratórios da Bell, Shannon aceitou o cargo de professor no MIT, atividade que exerceu durante muitos anos. Preocupava-se com os conceitos e simplificava ao máximo a simbologia. Onde outros professores poriam símbolos e mais símbolos, índices e mais índices, Shannon colocava duas ou três letras e incentivava os alunos a perceber as relações matemáticas que essas letras traduziam.

Gênio matemático que combinava a intuição, a abstração e as aplicações, Claude Shannon tinha como passatempos andar de monociclo, construir máquinas de jogar xadrez e outras aparentemente inúteis[†]. Estendeu sua Teoria Matemática de Comunicação ao campo da criptologia.

Claude Shannon, tinha a doença de Alzheimer. Faleceu no sábado, 24 de Fevereiro de 2001, no Courtyard Nursing Care Center em Medford, Massachusetts. Ele estava com 84 anos.

[*] Informalmente falando, trata da representação matemática das condições e parâmetros que afetam a transmissão e processamento da informação. É importante notar que "informação", como entendida na teoria da informação, não tem nada a ver com o significado inerente na mensagem. Significa um certo grau de ordem, de não randomicidade, que pode ser medida e tratada matematicamente como as quantidades físicas.

[†] Elaborou um autômato que procurava a saída num labirinto e aquela a que chamou de "máquina final". Nela, via-se apenas um interruptor. Ligando-o, o aparelho emitia um som zangado e dele emergia uma mão mecânica que desligava o interruptor, terminando a brincadeira.

6 As primeiras máquinas

6.1 Os primeiros computadores eletromecânicos

A partir da década de 1930 alguns cientistas começaram a trabalhar com dispositivos de cálculo com algum tipo de sistema de controle automático. Já se dispunha da tecnologia necessária para se construir aquela estrutura imaginada por Babbage. Surgiram os primeiros computadores mecânicos e eletromecânicos e muitos projetos de computadores eletrônicos feitos posteriormente sofreram muitas influências dessas primeiras máquinas.

6.1.1 Konrad Zuse

Figura 30: Konrad Zuse por volta dos anos 70

Konrad Zuse (1910 - 1995) foi o primeiro a desenvolver máquinas de cálculo controladas automaticamente. Esse engenheiro civil percebeu rapidamente que um dos aspectos mais onerosos ao se fazerem longos cálculos com dispositivos mecânicos era guardar os resultados intermediários para depois utilizá-los nos lugares apropriados nos passos seguintes **[Zus80]**. Em 1934, depois de várias idéias e tentativas, Zuse chegou à conclusão que um calculador automático somente necessitaria de três unidades básicas: uma controladora, uma memória e um dispositivo de cálculo para a aritmética. Ele desenvolveu o seu Z1, em 1936, um computador construído inteiramente com peças mecânicas e que usava uma fita de película cinematográfica para as instruções que controlavam a máquina.

Em 1938, antes mesmo de terminar o Z1, um aluno de Zuse, Helmut Schreyer, construiu uma parte do Z1 usando válvulas. Em função da situação de pré-guerra, Zuse teve

de abandonar essa linha de desenvolvimento – seriam necessárias 1000 válvulas, o que era impossível naquele momento – e continuou o Z2 usando tecnologia baseada em relês.

Esses dois primeiros modelos eram somente para teste: "tinham todas as características do computador posterior, mas não trabalhavam satisfatoriamente. O Z3 foi terminado em 1941 e foi o primeiro modelo totalmente operacional"* **[Zus80]**. O Z3, como a maioria das máquinas dessa primeira geração, usava dois mecanismos separados para as funções aritméticas e tinha uma unidade especial para conversão de números na notação decimal para a binária. Em termos de velocidade podia ser comparado ao MARK I, discutido mais à frente, que foi terminado dois anos após o Z3. O Z3 executava três a quatro adições por segundo e multiplicava dois números em quatro ou cinco segundos. Nunca chegou a ser usado para grandes problemas em função de possuir uma memória de tamanho limitado. Foi destruído, junto com a casa de Zuse, por um bombardeio em 1944.

O Z4 começou a ser desenvolvido quase que simultaneamente ao final do trabalho do Z3. Era essencialmente a mesma máquina, com maior capacidade de memória e mais rápida. Por causa do avanço das tropas aliadas, o trabalho do Z4 foi interrompido quase ao seu final e a máquina ficou escondida em uma pequena cidade da Bavária chamada Hinterstein Em 1950, na Suíça, Zuse reconstruiu o seu Z4, e fundou uma empresa de computadores, absorvida depois pela Siemens. As máquinas de Zuse tiveram pouco impacto no desenvolvimento geral da Computação pelo absoluto desconhecimento delas até um pouco depois da guerra **[Zus80]**.

6.1.2 As máquinas da Bell e as máquinas de Harvard

Por volta de 1937, enquanto Turing desenvolvia a idéia da sua "Máquina Universal" e formalizava o conceito do que é computar e do que é um algoritmo, nos Estados Unidos dois outros matemáticos também consideravam o problema da computação: Howard Aiken, em Harvard, cujo trabalho daria seus frutos em 1944, e George Stibitz, nos laboratórios da Bell Telephones. Eles procuravam componentes eletromecânicos que pudessem ser usados na computação de cálculos.

Nos últimos anos da década de 1930 os problemas envolvendo cálculos com números complexos (aqueles com partes imaginárias, envolvendo raízes negativas) no projeto de equipamentos telefônicos começaram a dificultar o crescimento da Cia. Bell Telephone. As pesquisas da empresa então começaram a ser direcionadas à descoberta de mecanismos que pudessem satisfazer essa necessidade cada vez mais crescente de cálculos mais rápidos. Stibitz demonstrou que relês podiam ser utilizados para executar operações aritméticas. A partir de 1938, juntamente com S. B. Willians começou a implementar suas idéias, e em 1939 estava pronto o seu Modelo I. Seus outros 'Modelos' chegaram até o número VI, terminado em 1950 – tendo estado em uso até 1961 –, e juntamente com os computadores K do Dr. Zuse foram os primeiros computadores de código binário, baseados em relês **[Sti80]**.

Ao mesmo tempo, nos Laboratórios de Computação de Harvard, Howard Aiken e engenheiros da IBM começaram a desenvolver um outro tipo de máquinas eletromecânicas,

* Quer dizer, tinha o controle automático das suas operações.

não totalmente baseada nos relês, já incorporando uma nova tecnologia que seria amplamente utilizada mais tarde: as memórias de núcleo de ferrite. Ao término de sua primeira versão, em 1943, o IBM Automatic Sequence Controlled Calculator, comumente chamado de Harvard Mark I*, tinha uma série de novas capacidades: modificava instruções dinamicamente baseando-se nos resultados obtidos durante o processamento, possuía unidades para decidir qual o melhor algoritmo para execução de um cálculo através do argumento de uma função, testava o conteúdo de registradores, etc. Diferenciava-se fundamentalmente de máquinas anteriores, como o EDSAC, (citado mais à frente), por usar memórias separadas para instruções e dados, o que ficou denominado como *arquitetura de harvard*. Quando terminado em 1944, foi imediatamente adotado pela marinha americana, para fins militares. Novas versões foram produzidas até 1952.

6.1.3 A participação da IBM

As máquinas de calcular mecânicas produzidas por empresas como a IBM não tinham linha de produção até a entrada dessas empresas no mercado de computadores propriamente dito. Eram equipamentos para auxiliar tarefas computacionais, que variavam desde as tabuladoras de Hollerith até tabuladoras para cálculos científicos como as produzidas por L.J. Comrie na Inglaterra ou Wallace J. Eckert nos Estados Unidos. Em 1935 a IBM começou a produzir suas séries 602, 602A até 605, calculadoras baseadas em relês que produziam em altas velocidades – como Babbage imaginou – tabelas de vários tipos, com alta confiabilidade. Uma posterior evolução foi a possibilidade dessas máquinas poderem ser programadas através de painéis de controle ('plugboards') para ler um cartão, executar até 60 diferentes cálculos aritméticos, e perfurar o resultados no próprio cartão de leitura. Outras empresas como a Remington Rand produziram equipamentos semelhantes **[Wil97]**.

Depois do sucesso do Mark I em Harvard, no qual teve grande participação com o laboratório em Endcott, a IBM lançou-se na produção do Selective Sequence Electronic Calculator (SSEC), sob o comando de Frank Hamilton, que pertenceu ao grupo de Aiken, em Harvard. Terminado em 1947, atraiu um importante grupo de pesquisadores que buscavam o aprimoramento da capacidade de cálculo e cujas soluções apontavam para um conceito decisivo para os computadores: o de *programa armazenado*. O último computador eletromecânico produzido foi o CPC, Card-Programmed Electronic Calculator, modelos I e II que finalizaram a série 700 **[Hur80]**.

6.2 O início da era da computação eletrônica

Durante os anos de 1936 a 1939, "John Vincent Atanasoff, com John Berry, desenvolveu a máquina que agora é chamada de **ABC** (Atanasoff-Berry Computer), na Universidade de Iowa, EUA, como uma máquina dedicada especialmente à solução de conjuntos de equações lineares na Física. Embora sendo um dos primeiros exemplos de

* Não confundir com um protótipo do computador EDASC, construído na Universidade de Manchester em 1948, baseado no conceito de programa armazenado

calculadora eletrônica, o ABC propiciou o desenvolvimento dos primeiros conceitos que iriam aparecer nos computadores modernos: a unidade aritmética eletrônica e a memória de leitura e gravação" **[IEEE95]**.

6.2.1 Estados Unidos: ENIAC, EDVAC e EDSAC

No início da Segunda Guerra, as necessidades de melhores tabelas de cálculo para as trajetórias de tiros tornaram-se imperativas, pois os analisadores diferenciais estavam no seu limite. Nessa época foi então montado, na Moore School of Electrical Engineering, da Universidade da Pensilvânia (Filadelfia, EUA), um grupo de pesquisa para o desenvolvimento de projetos eletrônicos relacionados ao futuro radar. Entre eles, J. Presper Eckert, Joseph Chedaker e Kite Sharpless logo se envolveram na produção de circuitos eletrônicos usados como contadores.

Eckert (1919-1995) e um pouco mais tarde John Mauchly (1907-1980), físico, e Herman H. Goldstine, matemático, acabaram por tornarem-se os principais protagonistas na construção do primeiro computador de uso geral que realmente funcionou como tal, o ENIAC (Electronic Numerical Integrator and Computer): esta máquina e a equipe que a projetou e construiu, serão responsáveis por um grande salto no desenvolvimento dos computadores eletrônicos. Seu formato era em U, suas memórias tinham 80 pés de comprimento por 8,5 de largura, e cada um dos seus registradores de 10 dígitos media 2 pés. Ao todo possuía 18.000 válvulas. Executava desvios condicionais e era programável, o que o diferenciava das outras máquinas construídas até a data. Sua programação era feita manualmente, através de fios e chaves. Os dados a serem processados entravam via cartão perfurado. Os programas típicos do ENIAC demoravam de meia hora a um dia inteiro para serem elaborados e executados.

Em 1944 John von Neumann ingressou como consultor na equipe da Universidade da Pensilvânia. Os responsáveis pelo projeto estavam interessados em melhorar a maneira como os programas eram desenvolvidos e iniciaram discussões a respeito do armazenamento de programas na forma de números. Iniciaram assim um trabalho sobre projetos de computadores que foi fundamental nos 40 anos que se seguiram. Em 30 de junho de 1945, von Neumamm publicou o *First Draft of a Report on the EDIVAC* (Electronic Discrete Variable Automatic Computer), que estabeleceu o paradigma de projetos de computadores para várias gerações seguintes de máquinas. Essa arquitetura ficou conhecida com o nome de "arquitetura de von Neumann", e entre outras coisas incluía o conceito de *programa armazenado*[*]. O ENIAC começou a operar em 1943, tendo sido terminado totalmente em 1946, encerrando suas operações em 1955. A saída dos professores Eckert e Mauchly da equipe atrasou, no entanto, o desenvolvimento do projeto EDVAC, só concluído em 1952.

Em 1946, Maurice Wilkes, da Universidade de Cambridge, visitou a Moore School para participar de uma conferência sobre computadores. Ao regressar a Cambridge, decidiu iniciar um projeto para um computador baseado no princípio do programa armazenado,

[*] Muitos dos pioneiros do desenvolvimento dos computadores acreditam que esse termo dá um crédito exagerado ao trabalho de von Neumann, que escreveu as idéias, e muito pouco aos engenheiros Eckert e Mauchly, que construíram as máquinas. A polêmica foi ruidosa e, em 1947, estes dois últimos deixaram a Moore School.

chamado EDSAC (Electronic Delay Storage Automatic Calculator), que se tornou operacional em 1949. Foi o primeiro computador de grande porte, baseado no citado conceito, que entrou em operação.

6.2.2 A contribuição inglesa: o COLOSSUS

Enquanto isso, na Inglaterra, o grande esforço era decifrar o código secreto de guerra germânico. Um grupo especial formado por cientistas e matemáticos reuniu-se em Bletchley Park, um lugar entre as Universidades de Cambridge e Oxford, para tentar construir uma máquina capaz de decodificar o alfabeto produzido pela versão germânica de um dispositivo inventado nos Estados Unidos, o ENIGMA **[Kah67]**. A equipe era liderada pelo prof. T. H. Flowers, sendo o prof. M. H. A. Newman o responsável pelos requisitos que levariam, em 1943, à construção do computador digital eletrônico **COLOSSUS*.** O trabalho do grupo de Betchley foi enormemente influenciado pelos resultados sobre computabilidade obtidos por Alan Turing. Este trabalhava no departamento de comunicação criado pelo governo britânico em função da guerra, com a missão de treinamento em lógica matemática e logo alocado também para os esforços de decifrar o código secreto alemão. O COLOSSUS acabou não sendo conhecido em sua época por duas razões. A primeira é não ter sido ele um computador para uso geral, mas sim projetado especialmente para decodificar mensagens secretas. A segunda: a existência desta máquina somente foi revelada a partir de 1970, sendo que seus algoritmos de decodificação são ainda secretos.

Ainda na Inglaterra, após o fim da guerra, Turing uniu-se ao centro de pesquisas do National Physical Laboratory, onde rapidamente elaborou o projeto básico do Automatic Computing Engine (ACE), que iniciou operações em 1950.

6.2.3 Outras contribuições

Um importante aspecto do desenvolvimento dos computadores foi a produção de dispositivos chamados de 'memória'. Desde Konrad Zuse, a construção de computadores que pudessem ter seus programas armazenados preocupou os cientistas e foi um fator determinante nas primeiras arquiteturas. Das memórias mecânicas de Zuse, passando pelas memórias térmicas – utilizadas somente experimentalmente – de A. D. Booth, pelos sistemas de *linha de retardo* baseados em mercúrio de Willian Shockley, da Bell – aperfeiçoada por Presper Eckert – utilizados no ENIAC, pelas memórias eletrostáticas de Willians até os núcleos magnéticos de ferrite, um árduo caminho foi percorrido. A memória de núcleos magnéticos acabou preponderando, tendo sido utilizada primeiramente em uma máquina de teste no MIT e mais tarde no computador conhecido como Whirlwind. O uso das memórias de núcleo magnético aumentaram excepcionalmente o desempenho dos computadores, podendo ser consideradas como um divisor de águas no desenvolvimento dos mesmos.

Importantes também nesse primeiro período foram duas grandes revoluções tecnológicas: o emprego de válvulas para tornar o computador mais rápido, confiável e de uso geral, e o conceito de programa armazenado. Esta técnica de usar uma "memória de

---

* Tornado operacional em 1944, decodificando mensagens para ajudar nos planos do desembarque do dia D, ainda nesse ano.

armazenamento" assim como a "transferência de controle via condição", que permitiam parar e reiniciar o processamento a qualquer instante, abriu enorme perspectiva para a programação de computadores. O elemento chave dessa arquitetura era a unidade central de processamento, que permitia a coordenação de todas as funções do computador através de uma única fonte. Em 1951, o UNIVAC I (Universal Automatic Calculator), construído pela Remington-Rand, tornou-se o primeiro computador *comercialmente disponível* que utilizava esses conceitos.

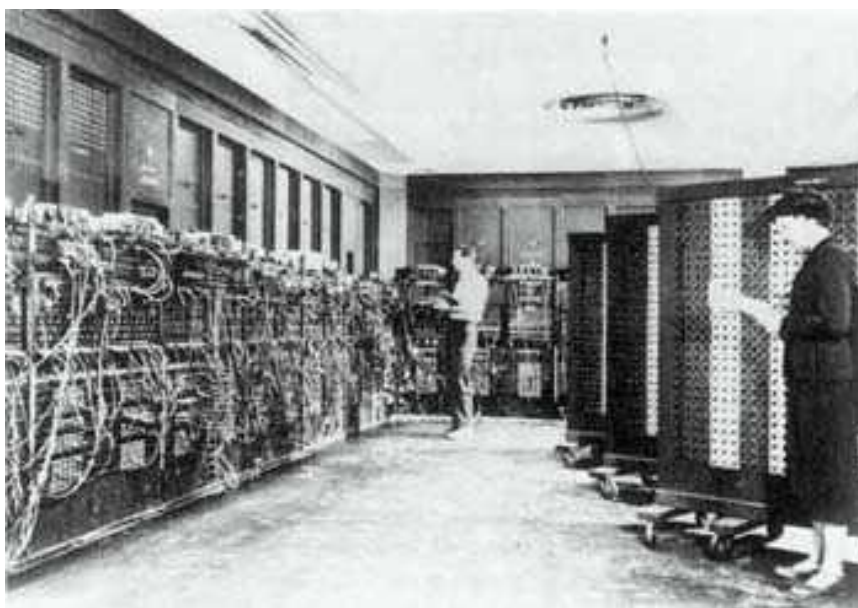



Figura 31: ENIAC, sua programação era feita com fios ("hard wired")

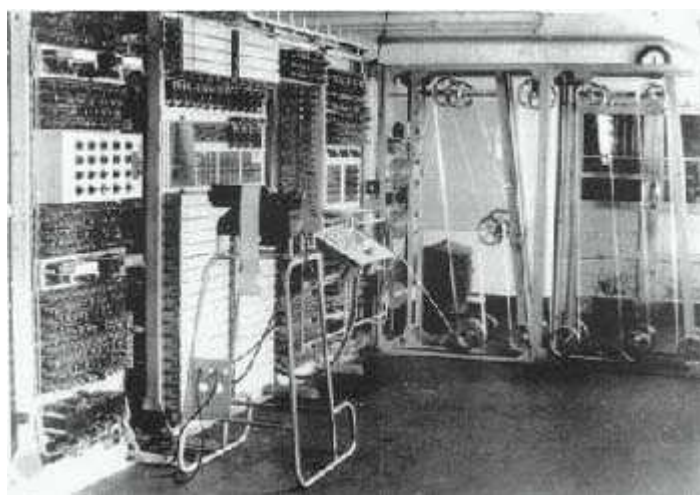



Figura 32:Colossus, da Inglaterra. Sua programação também era feita com fios.

Esta primeira geração de computadores caracterizou-se pelo fato de que as instruções de operação eram produzidas para tarefas específicas. Cada máquina tinha um

programa em código binário diferente que indicava o fluxo das operações. Isto dificultava a programação e limitava a versatilidade desses primeiros computadores.

Figura 33:da esquerda para a direita, Patsy Simmers, segurando uma placa do ENIAC, Gail Taylor, segurando uma placa do EDVAC, Milly Beck, segurando uma placa do ORDVAC, Norma Stec, segurando uma placa do BRLESC-I (atenção para o tamanho das placas)

Na figura que se segue há um pequeno resumo desses primeiros tempos da computação eletrônica, tomando como referência o ensaio de Arthur W. Burks **[Bur51b]**.

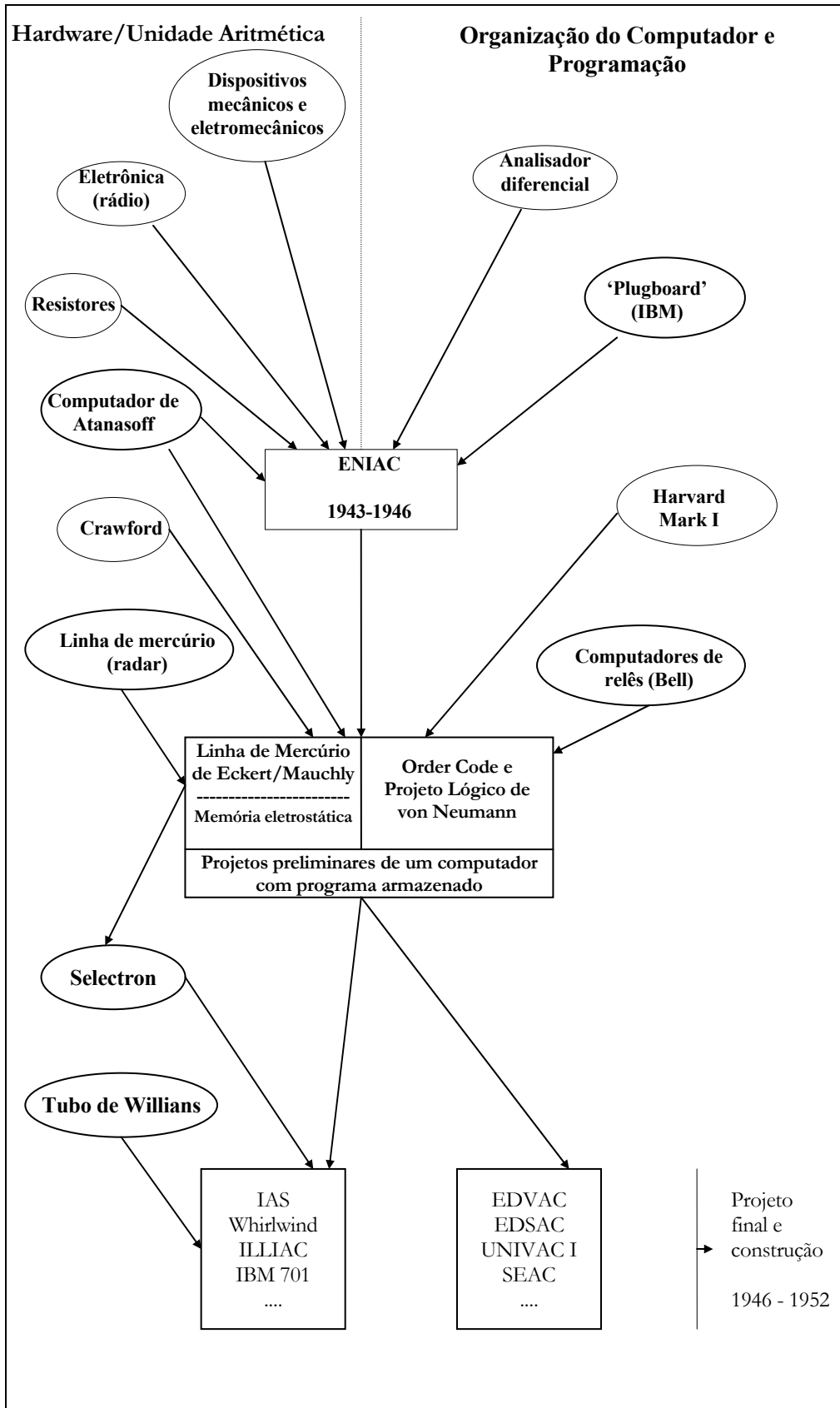



Figura 34: Desenvolvimento do hardware e software nos primeiros tempos da Computação

6.3 As primeiras linguagens

Seguindo as premissas iniciais – de se destacar neste trabalho a evolução dos conceitos e idéias ao longo da História da Computação – não se fará aqui uma descrição exaustiva da evolução das linguagens de programação. Depois de algumas considerações teóricas iniciais, serão vistas as motivações e as primeiras tentativas de se estabelecer um código que pudesse ser lido e processado pelos primeiros computadores.

6.3.1 Alguns aspectos teóricos

Como foi visto, um dos pontos fundamentais do projeto formalista de Hilbert para a solução de problemas matemáticos era descobrir um procedimento efetivo (ou mecânico) para verificar a validade de proposições matemáticas. Depois do Teorema de Gödel evidenciou-se que tal proposta é irrealizável, mas todos os estudos em torno desse projeto de Hilbert e dos resultados de Gödel propiciaram, entre outras coisas, uma adequada caracterização do termo *efetivamente computável*, através da Máquina de Turing e das funções lambda-definíveis de Church e Kleene. Tornou-se claro o que é um procedimento efetivo, tornando-se claro ao mesmo tempo o que é um problema computável.

Um procedimento efetivo é uma seqüência finita de instruções que podem ser executadas por um agente computacional, seja ele homem ou não. Propriedades:

I. a descrição deve ser finita;
II. parte de um certo número de dados, pertencente a conjuntos específicos de objetos, e espera-se que produza um certo número de resultados que mantenham relação específica com os dados;
III. supõe-se que exista um agente computacional – humano, eletrônico, mecânico, etc. – que execute as instruções do procedimento;
IV. cada instrução deve ser bem definida;
V. as instruções devem ser tão simples que poderiam ser executadas por alguém usando lápis e papel, em um espaço de tempo finito.

Esse procedimento efetivo também é chamado de *algoritmo*. Programas de computadores que terminam sua execução, fornecido qualquer conjunto específico de dados de entrada, são algoritmos.

A descrição finita do algoritmo deve ser feita através de uma determinada *linguagem*. Essa linguagem algorítmica deve pertencer a um subconjunto não ambíguo de uma linguagem natural, tal como Francês ou Inglês, ou ser uma linguagem artificial construída para isso, como, por exemplo, as linguagens de programação (Fortran, Lisp, Ada, Cobol, Pascal, etc.). As frases da linguagem descreverão as operações a serem executadas. A forma ou formato de procedimentos efetivos em uma linguagem algorítmica qualquer é especificada por um conjunto de regras chamado *regras de sintaxe*, cujas propriedades estão acima enumeradas **[BL74]**. Essa sintaxe refere-se aos programas corretamente escritos nela e o relacionamento

entre os símbolos e frases que ocorrem nesses programas. Alguns autores a dividem em *concreta* e *abstrata* **[Mos92]**. A concreta envolve:

- reconhecimento de textos (seqüências de caracteres) corretamente escritos de acordo com as especificações da linguagem;
- a colocação dos textos, de maneira não ambígua, dentro das frases que compõe o programa. A sintaxe abstrata molda as estruturas de frases do programa.

Portanto a sintaxe refere-se à forma dos programas: de que modo expressões, comandos e declarações podem ser justapostos para compor um programa. Uma linguagem de programação torna-se assim, entre outras coisas, uma notação formal para a descrição de um algoritmo, entendendo-se por *notação formal* um simbolismo que não tenha as imprecisões nem a variabilidade de uma linguagem natural, que possibilite rigor nas definições e demonstrações sobre os procedimentos.

Uma linguagem de programação necessita ainda de outros requisitos. Deve ser universal, isto é, que qualquer problema *cuja solução possa ser encontrada através de um computador* pode ser escrito com ela*. Na prática deve ser apta a resolver, no mínimo, os problemas da área para a qual foi projetada†. E uma característica fundamental: ser implementável em computador, isto é, deve ser possível executar qualquer procedimento bem formado na linguagem **[LSSK79]**.

Uma linguagem de programação também possui uma *semântica*. A semântica de um programa irá depender exclusivamente do que se deseja *causar*‡ objetivamente quando o programa for executado por um agente computacional, eletrônico ou não. Os computadores atualmente são máquinas complexas. Quando estão executando programas, luzes se acendem, cabeçotes dos discos movem-se, corrente elétrica flui pelos circuitos, letras aparecem na tela ou são impressas, e assim por diante. Um programa controla todos esses 'fenômenos' mediante sua semântica. E se são consideradas as linguagens de programação de *alto nível*, que não controlam diretamente esses detalhes de ordem física, falar de semântica significa falar das características que tornam tais linguagens implementáveis em qualquer computador, isto é, quais as características da execução do programa que são comuns a todas as implementações. Portanto a semântica é uma entidade abstrata: ela modela o que o programa quer causar quando é executado, independentemente do seu uso nesse ou naquele computador. A semântica de uma linguagem de programação é a mesma semântica de todos os programas escritos nela **[Mos92]**.

A evolução das linguagens de programação chegou até esses conceitos por caminhos e esforços muitas vezes paralelos. Alguns informatas buscam caminhos para projetar linguagens que combinem uma grande generalidade de usos (aplicações matemáticas e científicas, gráficas, comerciais, etc.) com simplicidade e eficiência. Isso levou ao desenvolvimento de diferentes paradigmas – estilos e objetivos – de programação como o *imperativo, o funcional, o orientado a objeto, o lógico, etc.* Outros buscaram e buscam caminhos para

---

* Qualquer linguagem em que se possa definir uma função recursiva será universal.

† Uma linguagem com somente tipos numéricos e arrays deve resolver naturalmente problemas numéricos, por exemplo.

‡ A maioria dos livros ao falar de semântica usa a palavra *behavior*, de difícil tradução. Pode-se dizer que é um conjunto de regras que determinam a ordem na qual as operações do programa irão ser executadas, quais serão executadas primeiro e quando se encerrarão.

expressar a sintaxe e a semântica, esta última talvez a parte mais importante dentro do assunto linguagens de programação e que levou ao surgimento de diversas linhas: a semântica algébrica, a denotacional, a de ações, etc.

6.3.2 Desenvolvimentos anteriores a 1940

Mas, antes de entrar nesse mundo das linguagens, de que forma eram anteriormente especificados os algoritmos? Os mais antigos algoritmos escritos que se conhecem são os da velha Mesopotâmia. Eram seqüências de cálculos sobre conjuntos particulares de dados e não uma abstração* de procedimento como entendido na programação atual **[Knu76]**. Na civilização grega, vários algoritmos não triviais foram estudados, como por exemplo o de Euclides. A descrição era ainda informal.

A notação matemática começou a evoluir efetivamente a partir dos séculos XIII e XIV e notações para relações funcionais tiveram um bom desenvolvimento. Na Computação, Babbage e Lady Lovelace elaboraram, entre outros, um programa para o cálculo dos números de Bernoulli **[Mor61].** Era na verdade uma espécie de programa em linguagem de máquina, como nos primórdios dos computadores digitais na década de 1940.

Em 1914, Leonardo Torres e Quevedo usaram uma linguagem natural para descrever um pequeno programa para seu autômato hipotético. Helmut Schreeyer fez uma descrição análoga em 1939 para a máquina que construía juntamente com Zuse.

O próprio Alan M. Turing, para tratar do problema da indecidibilidade de Hilbert construiu uma linguagem muito primitiva para sua máquina. Nela só havia comandos para ler, testar uma condição e escrever símbolos sobre uma fita, movendo para a direita ou esquerda uma cabeça de leitura e gravação. Conforme Knuth **[KP80]**, as 'tabelas' de Turing (como Alan Turing chamava sua linguagem) "representaram a notação de mais alto nível para uma descrição precisa de algoritmo que foram desenvolvidas antes da nossa história começar – exceto talvez pela notação-lambda de Alonzo Church (que representa um 'approach' inteiramente diferente para o cálculo) ".

6.3.3 As primeiras tentativas

Nos primeiros tempos da computação propriamente dita os programas eram escritos em código de máquina e colocados diretamente no computador por meio de cabos e fios. Por exemplo:

0000 0001 0110 1110
0100 0000 0001 0010
1100 0000 0000 1101

* "Uma *abstração* é um modo de pensar pelo qual nos concentramos em idéias gerais ao invés das manifestações específicas destas idéias.(...) Na programação, a abstração refere-se à distinção que fazemos entre: (a) *o que* um pedaço de programa faz e (b) *como* ele é implementado. Uma linguagem de programação em sentido próprio consiste de construções que são (em última instância) abstrações do código de máquina" **[Wat90].** Exemplos típicos de abstrações são as funções e procedimentos de uma linguagem de programação.

Segundo Grace Murray Hopper[*], como curiosidade, "a frase mais freqüente que nós ouvíamos era que a única maneira de se programar em um computador era em octal" **[Wex80]**. Em 1946, junto com Howard Aiken, era assim que ela programava o Mark I.

Percebeu-se claramente que os programas em código de máquina eram extremamente difíceis de editar e modificar, e quase impossíveis de se compreender. A comunidade computacional logo entendeu que era necessário inventar uma notação simbólica para tornar os programas mais fáceis de escrever. Nesta evolução as instruções acima ficam com o formato:

LOAD X
ADD R1 R2
JUMPZ H

Uma vez feito o programa dessa maneira, o programador o prepararia para ser executado, 'escrevendo' manualmente (em painéis, através de um emaranhado de cabos e 'plugs') as instruções no correspondente código de máquina. Este processo foi chamado de *assembling*. O que depois se queria fazer era com que a própria máquina executasse essa operação.

Mas mesmo quando programava com esses códigos de operação mnemônicos (também chamados de linguagem de montagem), o programador ainda estava trabalhando em termos dos conjuntos de instruções da máquina, isto é, os algoritmos eram expressos em termos de instruções muito primitivas (detalhes sobre registradores, endereços, saltos, etc.). Daí a denominação *linguagens de baixo nível*. A busca de linguagens que pudessem permitir que os algoritmos fossem expressos em termos análogos à idéia elaborada na mente do programador fez com que aparecessem os primeiros compiladores e começassem a surgir as chamadas linguagens de *alto nível*[†].

Claramente percebem-se duas principais tendências nesses anos pioneiros: aqueles que procuravam saber o que era possível implementar e os que estavam preocupados com o que era possível escrever. Estes últimos criaram estruturas conceituais – iteração, tipos de dados, recursividade, etc. – importantes no processo de programação e que foram depois objetos de estudo na Teoria da Computação. Naturalmente foram precisos muitos anos para que essas duas tendências se juntassem para formar uma síntese adequada.

6.3.4 Konrad Zuse e seu 'Plancalculus'

Depois de salvar o Z4 das bombas dos aliados e mudar-se para a pequena vila Hintesrtein nos Alpes, Konrad Zuse percebeu que ainda não existia uma notação formal para a descrição de algoritmos e começou a trabalhar em uma. O resultado foi uma linguagem chamada *Plankalkül* (program calculus), uma extensão do cálculo proposicional e de

[*] Nome importante no desenvolvimento histórico das linguagens de programação. Ela desenvolveu programas para o Mark I, um dos precursores do computador moderno; esteve envolvida na construção do UNIVAC e trabalhou no primeiro compilador que se tem notícia, o A-2, e em uma das primeiras linguagens matemáticas, originalmente chamada A-3 e depois MATH-MATIC. Em 1955 trabalhou na equipe que elaborou as primeiras especificações para uma linguagem de uso comercial, originalmente chamada B-0, depois FLOW-MATIC, que forneceu inúmeras características para o COBOL.

[†] O termo 'alto nível' refere-se à semelhança que a linguagem tem com uma linguagem natural ou matemática, opondo-se a 'baixo nível', mais semelhante à linguagem de máquina.

predicado de Hilbert. Em uma monografia sobre o Plankalkül, em 1945, Zuse começava dizendo: "A missão do Plancalculus é fornecer uma descrição formal pura de qualquer procedimento computacional". O Plancalculus incluía alguns conceitos fundamentais da programação: tipos de dados, estrutura de dados hierárquicos, atribuição, iteração, etc. Ele pensou inclusive em usar o Plancalculus como base de uma linguagem de programação que pudesse ser traduzida por uma máquina. "Pode-se resumir sua idéia dizendo que o Plankalkül incorporou muitas idéias extremamente importantes, mas faltou-lhe uma sintaxe amigável para expressar programas em um formato legível e facilmente editável". Como complementação de seu trabalho desenvolveu algoritmos para ordenação, teste de conectividade de grafos, para aritmética de inteiros (inclusive raiz quadrada) e até um jogo de xadrez, entre outros. Infelizmente a maior parte destas coisas permaneceu desconhecida até 1972, a não ser por alguns extratos aparecidos em 1948 e 1959, quando seu trabalho chamou a atenção de alguns leitores ingleses. "É interessante especular sobre o que teria acontecido se ele tivesse publicado tudo imediatamente; teriam as pessoas sido capazes de entender idéias tão radicais?" **[KP80].**

6.3.5 O diagrama de fluxos

Em Princeton, do outro lado do Atlântico, Herman H. Goldstine e John von Neumann estavam preocupados com o mesmo problema: como se poderiam representar algoritmos de uma maneira precisa, em uma linguagem de mais alto nível que a de máquina.

Ao desenvolver os projetos lógicos do computador EDVAC e da máquina do IAS (Institute for Advanced Study, da universidade de Princeton), von Neumann tinha também uma grande preocupação com a sua programação. Ele deixou um manuscrito que é provavelmente o primeiro programa escrito para um computador com programa armazenado na memória*. O problema proposto é o da classificação de uma série de dados em ordem crescente. Von Neumann propôs o método que ficou conhecido mais tarde como *classificação por intercalação*, até hoje um dos algoritmos mais usados para classificar dados na memória†. Já aí aparecem alguns expedientes que prenunciam o surgimento das linguagens de montagem (linguagens simbólicas, mnemônicas, muito próximas à linguagem binária das máquinas) como os símbolos para denotar grandezas ou o conceito básico de *alocação dinâmica de memória*‡. A descrição termina com uma análise do tempo de execução muito semelhante às análises difundidas por Donald Knuth.

---

* Uma história e análise desse manuscrito estão em um artigo feito por Knuth em 1970, intitulado *von Neumann's First Computer Program*

† Na realidade o manuscrito contém somente uma parte da codificação do método, que é a parte do processo de duas sequências já em ordem

‡ Sem entrar em detalhes mais técnicos, significava que a atribuição de endereços era feita em relação a um endereço inicial arbitrário, a ser preenchido mais tarde, conseguindo-se o efeito de relocação manual do código, de modo a ser usado como uma *subrotina aberta* (outro conceito que exige maior conhecimento técnico sobre sistemas operacionais e programação/arquitetura de computadores)

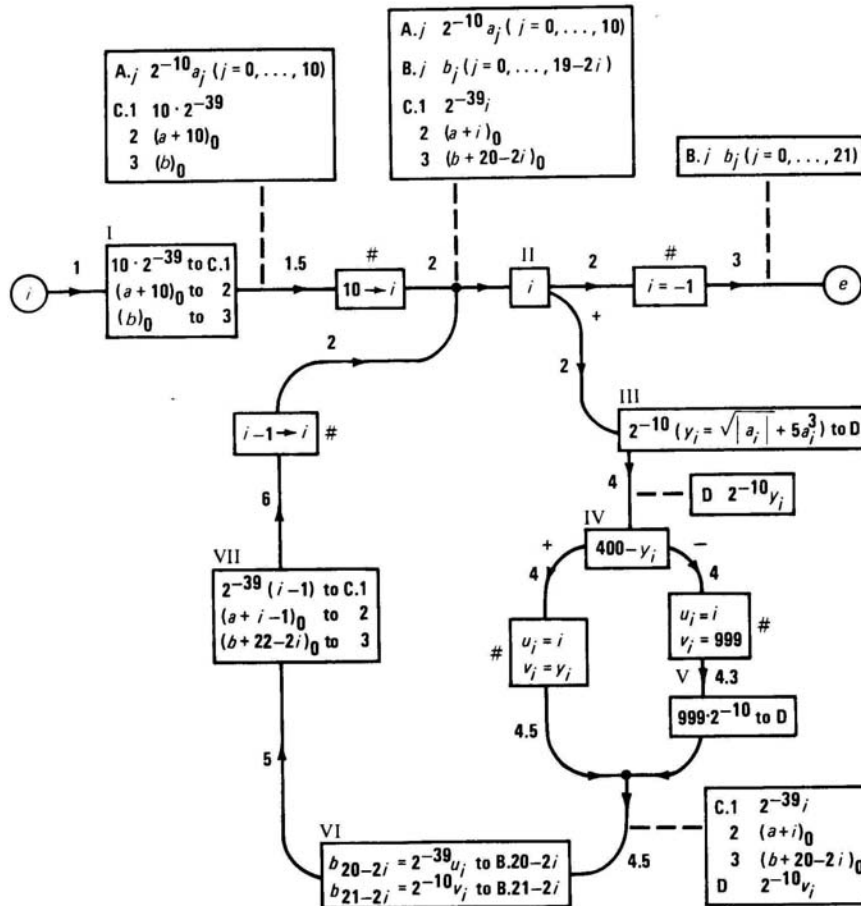



Figura 35: Um exemplo de um diagrama de fluxos

Para o projeto do IAS, von Neumann e Goldstine compuseram três volumes (1947/48) intitulados *Planning and coding of problems for an electronic computing instrument*. O primeiro volume é dedicado à metodologia da programação, o segundo volume traz vários exemplos de programas e o terceiro volume é dedicado à construção de subrotinas reutilizáveis e construção de bibliotecas destas subrotinas.

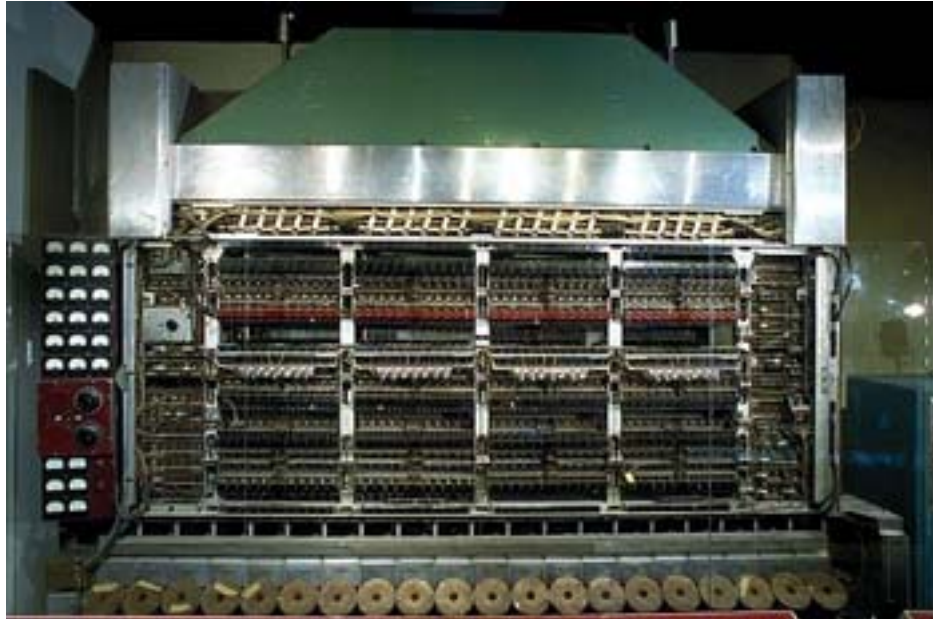



Figura 36: Computador IAS, 1952

Eles propuseram uma representação pictórica, através de caixas unidas por setas, que chamaram de fluxogramas. Descreveram fluxogramas que continham uma caixa denominada "caixa de anotação (especificação)". Nessa caixa descreviam-se certos fatos sobre o resultado de uma computação (o efeito por ela provocado). O conteúdo dessa caixa deveria ser confrontado com as operações descritas pelo fluxograma, possibilitando uma verificação da consistência entre o fluxograma e as intenções do programador expressas através das anotações. Com von Neumann e Goldstine encontra-se também a primeira referência à corretude de programas.

A ênfase era colocada no poder de cálculo – e não na expressividade das estruturas como Zuse – e esse trabalho foi largamente difundido entre as pessoas envolvidas com computadores na época. "Tal fato, acompanhado da excelente qualidade de apresentação e pelo prestígio de von Neumann, significaram que seu trabalho teve um enorme impacto, tornando-se fundamento para técnicas de programação em todo o mundo". O conceito matemático de igualdade foi substituído pelo de *atribuição* **[KP80]**.

6.3.6 A contribuição de Haskell

Haskell B. Curry, contemporâneo de Goldstine e von Neumann, após uma experiência com um programa complexo no ENIAC, sugeriu uma notação mais compacta que a deles. Na prática não obteve sucesso pela maneira estranha com que analisava e dividia os problemas. O principal ponto de interesse no trabalho de Curry, no entanto, não foi a sua linguagem de programação, mas os algoritmos que analisou para conversão de parte desses em código de máquina. Com isso proporcionou uma descrição recursiva de um procedimento para converter expressões aritméticas claras em um código de máquina apropriado, sendo por isso a primeira pessoa a descrever a fase de geração de código de um compilador **[KP80].**

6.4 Interpretadores algébricos e linguagens intermediárias

Em 1951, promovidas pela Marinha americana, houve uma série de três conferências, e naquela que tratava sobre manipulação de dados e computação automática apareceu um trabalho de John Mauchly, o Short Order Code. Codificado para o computador BINAC, em 1949, por William F. Schmitt, foi recodificado pelo mesmo em 1950 para o UNIVAC e usava dois dígitos para representar alguns símbolos, ao invés do usual código binário. Era na verdade uma espécie de *interpretador* algébrico: o programa percorria cada linha de representação de código, da direita para a esquerda, desviava para as chamadas subrotinas, executava-as e ia para a próxima instrução.

Arthur W. Burks e colegas, na Universidade de Michigan, investigando o processo de passar alguns problemas de processamento de dados descritos em uma linguagem comum para a 'linguagem interna' de um computador, esboçaram uma 'linguagem intermediária' que seria o passo anterior de uma 'linguagem interna' do computador, e que tinha um alto nível de abstração **[Bur51a]**.

Heinz Rutishauser**,** colaborador de Zuse no Z4, publicou em 1952 um trabalho descrevendo um computador hipotético e uma linguagem algébrica simples, junto com os fluxogramas de von Neumann para o que seriam dois 'compiladores'* para essa linguagem. Um para decodificar todos os *loops* enquanto o outro produzia código compacto através de registradores de índice. Como o Short Code, o programador deveria reservar manualmente as localizações de memória para variáveis e constantes. Um trabalho semelhante apareceu na Itália, na tese de dissertação de Corrado Böhm, que desenvolveu uma linguagem algébrica e o primeiro compilador para ela na própria linguagem, que reconhecia precedência de operações.

6.5 Os primeiros 'compiladores'

Conforme Knuth e Trabb **[KP80],** o termo compilador não era ainda utilizado nessa época. Na verdade falava-se sobre *programação automática*. No início da programação em linguagem de máquina foram desenvolvidas subrotinas de uso comum para entrada e saída, para aritmética de ponto flutuante e funções transcendentais. Junto com a idéia de um *endereçamento realocável* – pois tais subrotinas seriam usadas em diferentes partes de um programa – foram criadas rotinas de montagem para facilitar a tarefa de uso das subrotinas e de *endereçamento relativo*, idéia desenvolvida por Maurice V. Wilkes**.** Para isso foi inventada uma *pseudo linguagem de máquina*. Uma rotina *interpretativa* iria processar essas instruções, emulando um computador hipotético **[Gol72] [Knu69]**. Esse é o sentido do termo 'compilador' até aqui usado.

AUTOCODE foi o primeiro 'compilador' real, que tomava uma declaração algébrica e a traduzia em linguagem de máquina. Seu desconhecido autor, Alick E. Glennie, das forças armadas da Inglaterra, declarava em Cambridge, em 1953, sua motivação para elaborá-lo: "A dificuldade da programação tornou-se a principal dificuldade para o uso das

* Os termos interpretador e compilador na linguagem da computação têm um sentido técnico específico, que na época citada ainda não correspondiam ao atual significado.

máquinas. Aiken expressou sua opinião dizendo que a solução para esta dificuldade deveria ser buscada pela construção de uma máquina especial para codificar(...) Para tornar isso fácil deve-se elaborar um código compreensível. Tal coisa somente pode ser feita melhorando-se a notação da programação" **[KP80].** John Backus **[Wex80]** discute essa distinção que Knuth faz, citando J. Halcomb Laning, Jr. e Niel Zierler como os inventores do primeiro 'compilador' algébrico, para o computador Whirlwind. Como esta, são muitas as discussões ainda hoje sobre quem foi o pioneiro no assunto. De qualquer maneira esses primeiros sistemas denominados genericamente de *programação automática* (acima citada) eram muito lentos e não fizeram muito sucesso, embora tivessem sido fundamentais para preparar a base do desenvolvimento que se seguiu.

Este veio com o A-0, agora sim o primeiro compilador propriamente dito, desenvolvido por Grace Murray Hopper e equipe, aprimorado para A-1 e A-2 subseqüentemente. O próximo passo seria o A-3, desenvolvido em 1955, produzido ao mesmo tempo com o tradutor algébrico AT-3, mais tarde chamado MATH-MATIC.

Em 1952 a IBM construía o computador 701 e em 1953 foi montada uma equipe liderada por John Backus para desenvolver um código automático que facilitasse a programação. O resultado foi o Speedcoding. Backus tornou-se uma das principais figuras na história da evolução das linguagens de programação, tendo um papel fundamental no desenvolvimento dos grandes compiladores que viriam a partir do ano de 1955 como o FORTRAN e o ALGOL, além do estabelecimento da moderna notação formal para a descrição sintática de linguagens de programação, denominada BNF, Backus Normal Form.

6.6 A figura de von Neumann

Figura 37: John von Neumann

Assim como não se pode falar de Computabilidade, ou sobre a Teoria da Computação, ou ainda sobre os Fundamentos da Computação, sem deixar de citar nomes como Turing e Church, também em relação ao advento do computadores, a partir da década de 1940, não se pode deixar de falar da figura de János Louis von Neumann (1903 – 1957). Este húngaro conhecido como "Jancsi", uma forma diminutiva de seu prenome János (depois de naturalizado americano seria tratado pelo apelido Johnny), era no final de sua vida um dos mais poderosos homens em cena no comando da política americana com relação às ciências. Mas esta foi somente uma das muitas qualidades que o distinguiu nos diferentes países onde viveu e nos variados campos da inteligência em que exerceu sua atuação.

Tido como brilhante desde sua infância (falava grego com seu pai aos seis anos de idade **[Die81]**), entrou para a universidade de Budapeste em 1921 para estudar Matemática. Adotou a prática, pouco ortodoxa!, de ir à universidade somente no fim dos cursos, para prestar exames, saindo-se sempre muito bem. Enquanto isso, entre 1921 e 1923, na universidade de Berlin, estudou Química, tendo entrado em contato com nomes ilustres, como Albert Einstein, Fritz Haber e outros cientistas húngaros, como Denis Gabor, Leo Szilard e Eugene Wigner. Em 1923 dirigiu-se a Zurich, para estudar Engenharia Química, formando-se em 1925. Doutorou em Matemática, no ano de 1926, pela Universidade de Budapeste, com uma tese sobre a Teoria dos Conjuntos (com 20 anos, estimulado por um dos mais famosos matemáticos do século XX, David Hilbert, enunciou uma definição sobre números que ainda hoje é utilizada). Ainda fez um pós-doutorado na prestigiosa Universidade de Göttingen, onde estudou sob a direção de Hilbert, no período de 1926 a 1927. Tornou-se um dos membros da elite dos físicos revolucionários da mecânica quântica, e rapidamente ganhou grande reputação pelos seus trabalhos na Álgebra (como aluno de Hilbert foi uma das estrelas do formalismo), Mecânica Quântica e Teoria dos Conjuntos. Convidado nos anos 30 a visitar a universidade de Princeton, foi chamado para ser um dos seis professores de Matemática que formariam parte do recém-inaugurado Instituto de Pesquisas Avançadas (os outros eram J. W. Alexander, A. Einstein, M. Morse, O. Veblen, J von Neumann e H. Weyl), cargo que levou até o fim de sua vida.**[Asp90]**.

Conheceu Kurt Gödel (que como ele, naturalizou-se americano durante a 2ª guerra), e Church, orientador, naquela época, da tese de doutoramento de Turing. Tomou conhecimento da publicação deste, *On Computable Numbers with an Application to the Entscheidungsproblem*, e convidou-o para trabalhar como seu assistente, pois estava interessado em sua idéia, que envolvia os conceitos de um projeto lógico para uma máquina universal. Turing, no entanto, preferiu retornar a Cambridge e, um ano mais tarde, envolver-se-ia na construção do computador Colossus, em Bletchley Park, na Inglaterra.

O interesse de von Neumann por computadores encaminhou-se rapidamente por uma vertente diferente daquela seguida pelos seus colegas. Percebeu o potencial da nova máquina para solução matemática de problemas e não somente para elaboração de tabelas. Durante a guerra, os seus conhecimentos em hidrodinâmica, balística, meteorologia, teoria dos jogos, e estatística, foram colocados em uso em vários projetos. Esse trabalho levou-o a perceber que poderiam ser usados dispositivos mecânicos para computar cálculos e, embora se diga que seu primeiro envolvimento com um computador foi através do ENIAC, de fato nessa época ele estava com Howard Aiken, em Harvard, no projeto do Mark I (ASCC). Sua

correspondência em 1944 mostra seu interesse não somente pelo trabalho de Aiken, mas também com os computadores baseados em relês, de George Stibitz, e pelas pesquisas de Jan Schilt no Watson Scientific Computing Laboratory da Universidade de Columbia. No fim da II Guerra von Neumann tornou-se consultor, servindo a numerosos comitês com sua prodigiosa habilidade de rapidamente ver a solução de problemas.

Ele tinha uma grande capacidade de aglutinar ao seu redor cientistas muitas vezes separados por causa de exigências de segredo. Movia-se confortavelmente entre o pessoal de Los Alamos (National Laboratory e Manhattan Project) assim como entre os engenheiros da Moore School of Electrical Engineering, da Universidade de Pensilvânia, que estavam construindo o ENIAC. Uma combinação de diferentes desenvolvimentos científicos conduziram à invenção do ENIAC: novas tecnologia de válvulas, lógica booleana, as idéias de Babbage-Lovelace, as teorias de controle via retroalimentação (*feedback*), etc., e von Neumann era talvez o único que conhecia sobre todos estes temas, além de politicamente dar-se bem com as sociedades de Princeton, Los Alamos e Washington. No projeto Manhattan, trabalhou juntamente com Oppenheimer, Fermi, Teller, Bohr e Lawrenceand, que, entre outros, construíram a bomba atômica. **[Ula80] [Gol72]**.

6.6.1 O conceito de programa armazenado

Quando se terminou o ENIAC, era tarde para utilizar tal equipamento no esforço de guerra, mas certamente foi possível realizar o objetivo dos seus inventores: um cálculo balístico, que poderia tomar vinte horas de um especialista, seria agora feito pela máquina em menos de 30 segundos. Pela primeira vez a trajetória de um míssil poderia ser calculada em menos tempo do que levava o míssil real para atingir seu alvo. O primeiro problema a ser resolvido por essa máquina foi um ensaio de cálculo para a bomba de hidrogênio, então sendo projetada.

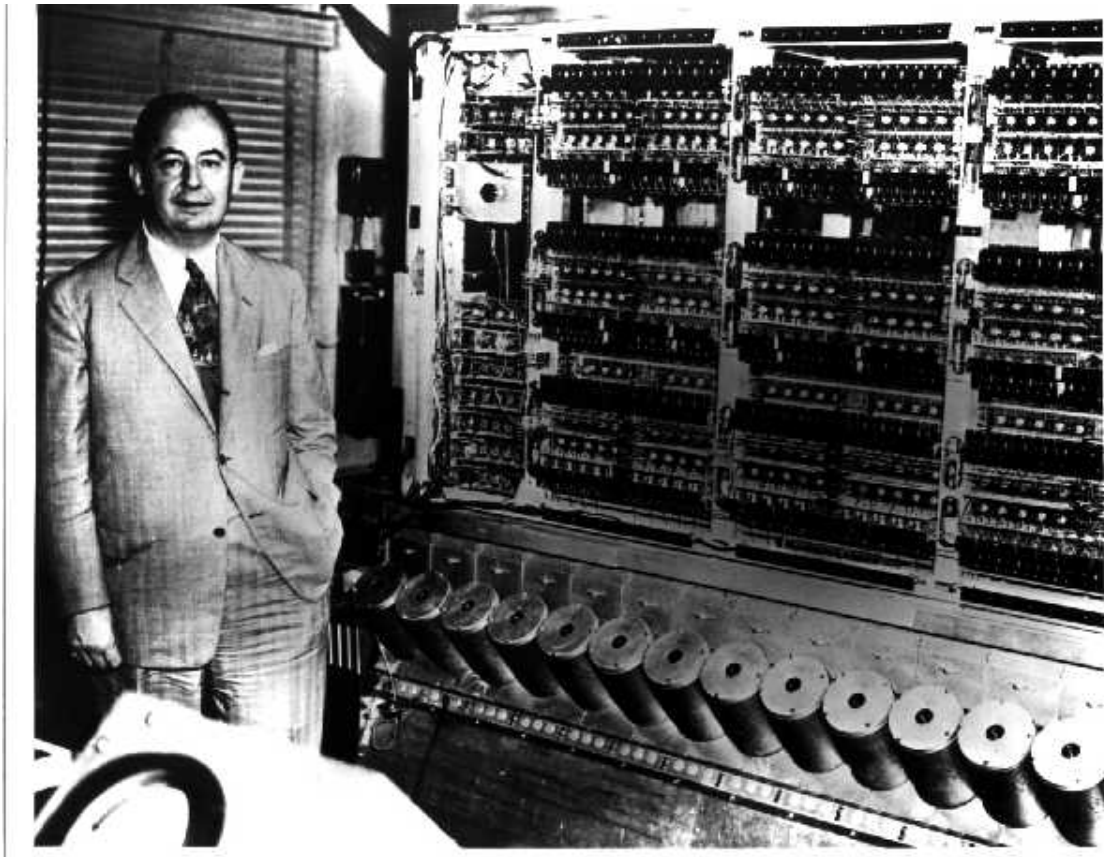



Figura 38: Von Neumann e o computador IAS

Von Neumann tinha se unido ao grupo em meados de 1944, através do matemático Herman H. Goldstine*, como consultor especial. Seu gênio para questões relacionadas ao pensamento formal, sistemático e lógico foi aplicado às propriedades daquela imensa máquina de 17.000 válvulas, 70.000 resistores e 10.000 capacitores. Os problemas relativos à 'engenharia' eram ainda imensos, mas estava se tornando claro que o componente não-físico, a *codificação*, aquilo que estabelecia a operação da máquina, era igualmente difícil e importante.

Até o aparecimento do transistor, o que ocorreu alguns poucos anos depois, o ENIAC representava o limite físico daquilo que poderia ser feito através de um grande número de chaves e conexões. Em 1945, o aprimoramento possível no poder computacional era um melhoramento na estrutura lógica da máquina, e von Neumann era provavelmente o único homem, a oeste da equipe inglesa de Bletchley Park (que tinha construído o computador COLOSSUS para decifrar o código germânico de guerra), preparado para compreender os mecanismos lógicos subjacentes no primeiro computador digital.

Parte da razão pela qual o ENIAC era capaz de operar rapidamente estava em que os caminhos seguidos pelos impulsos elétricos eram estabelecidos dentro do equipamento. Esta 'rota eletrônica' era a materialização das instruções de máquina que transformavam os dados de entrada em soluções de problemas. Diferentes tipos de equações poderiam ser resolvidas, e a performance dos cálculos poderia ser alterada pelos resultados de

* A partir desse momento deu-se o início de uma grande amizade entre os dois.

subproblemas*. Mas o que era ganho no poder de cálculo e velocidade era perdido na flexibilidade. Um sério obstáculo consistia na necessidade da programação externa, através de painéis e cabos de conexão para a solução de um determinado problema, um procedimento que poderia levar vários dias†. Após a entrada de von Neumann na equipe percebeu-se que o ENIAC não seria a última palavra em termos de máquinas calculadoras, mas sim que era o protótipo, ainda imperfeito, de uma nova categoria de máquinas. Antes mesmo de estar terminado, seus construtores já estavam elaborando o projeto de seu sucessor, e von Neumann compreendeu, daquelas discussões com seus colegas, que se estava falando de uma *máquina de uso geral*.

Na mesma época, a Universidade de Pensilvânia celebrou um contrato suplementar para a construção do EDVAC, proposta um pouco antes por Mauchly e Eckert, cujas características eram ainda um tanto vagas. O novo projeto despertou enorme interesse em von Neumann, que começou a participar de reuniões relativas ao projeto, juntamente com Eckert, Mauchly, Goldstine e outros.

Conforme **[Kow96]**, um fator decisivo para viabilizar o projeto de uma nova máquina foi a idéia de Eckert de utilizar *linhas de retardo* para implementar elementos de memória de custo muito mais baixo do que se fossem utilizadas válvulas. Outro resultado das reuniões com a equipe do projeto e da freqüente troca de correspondência, foi a produção de um documento descrevendo os detalhes da organização da nova máquina. Von Neumann ficou encarregado de escrever o *First draft of a report on the EDVAC* (1945), documento que nunca passou da fase de rascunho (foi publicado na íntegra somente anos mais tarde, com forma ligeiramente editada).

Existem controvérsias, alimentadas por nomes como Randell, Rosen, Stern e Wilkes, sobre quem teria sido o primeiro a propor o conceito de programa armazenado. O trabalho teórico de Turing, com o qual von Neumann estava familiarizado, já indicava essa possibilidade. Por outro lado existem referências bastante obscuras e ambíguas, em fontes anteriores ao documento de von Neumann, além das afirmações posteriores de Eckert e Mauchly. Não há dúvida de que a idéia de programa armazenado estava no ar, e é bastante provável que tenha sido sugerida por mais de uma pessoa. Apesar da notoriedade dessa controvérsia, não parece que sua importância seja mais que simbólica. Independentemente de quem tenha sido o primeiro a sugerir a idéia de programa armazenado na memória, o fato é que o documento redigido por von Neumann é a primeira descrição minuciosa e quase completa da arquitetura de um computador desse tipo, com o repertório de operações que permitiriam a utilização plena de seus recursos. Embora resultado de várias reuniões, o fato de von Neumann ter sido consultor no projeto e encarregado de sua redação indica o peso da sua contribuição. Depoimentos de colaboradores indicam que o projeto lógico do computador deve-se principalmente a ele, enquanto Eckert e Mauchly foram os principais

* O ENIAC estava habilitado de acordo com a idéia de Babbage, em seu Engenho Analítico, pela qual ele poderia ser reprogramado para solucionar diferentes equações não alterando a máquina, mas a seqüência dos cartões de entrada.

† A origem do ENIAC como um dispositivo voltado para um projeto balístico era parcialmente responsável por esta pouca flexibilidade. Não era a intenção dos engenheiros da Moore School construir uma máquina universal. O contrato estabelecido especificava claramente que eles deveriam criar um novo tipo de calculador de trajetórias.

responsáveis pelo projeto de circuitos de alta velocidade, linhas de retardo e outros detalhes físicos, contribuições igualmente fundamentais **[Kow96]**.

6.6.2 A arquitetura de von Neumann

O relatório de von Neumann ficou incompleto, mas sua leitura é instrutiva. Muitas idéias continuam válidas até hoje: a separação entre arquitetura lógica e física, a divisão do projeto em unidades de controle, aritmética, memória, entrada e saída, precursoras de todos os projetos posteriores. Além disso, devido ao interesse nos trabalhos relativos a sistemas neurais de McCulloch e Pitts, ele descreveu vários dispositivos do computador fazendo analogia com o sistema nervoso (mesmo porque na época não existia ainda uma linguagem adequada para tais descrições). Von Neumann, em um nível teórico, estava intrigado com algumas aparentes semelhanças na organização e funcionamento paralelo da mente e dos computadores, o que poderia levar a pensar em teorias lógico-formais que abrangessem tanto um quanto outro. De alguma maneira, pensava, esses mecanismos poderiam evoluir para algum tipo de extensão intelectual.

Com o fim da guerra em 1945, iniciaram-se gestões para construção de outro computador para aplicações científicas em geral, com o apoio do IAS e da RCA(Radio Corporation of America), assim como da Marinha e Exército. O projeto foi descrito em um documento básico, composto de duas partes, onde a primeira é a mais fundamental, referente ao projeto lógico. Conjuntamente com a descrição incompleta do EDVAC, esse esforço constituiu a inspiração para a elaboração da arquitetura que foi, e continua sendo, modelo de quase todos os projetos de computadores subsequentes:a arquitetura de von Neumann. A expressão parece ter sido usada pela primeira vez por J. Backus, em 1977, durante o recebimento do Prêmio Turing da ACM, em palestra intitulada *Can programming be liberated from the von Neumann style? A functional style and it's algebra of programs.* Nela criticava o fato de que após mais de 30 anos da sua introdução, as arquiteturas de von Neumann ainda eram dominantes e exerciam enorme influência sobre o paradigma imperativo das linguagens de programação mais utilizadas, impedindo o desenvolvimento de outros modelos*.

No entanto pode-se afirmar "(...) que a estrutura lógica introduzida nos projetos do EDVAC e da máquina IAS constitui o princípio de funcionamento de computadores digitais até hoje, apesar do progresso tecnológico que nos separa daquela época. Na realidade, não parece provável que os conceitos básicos de arquitetura de von Neumann sejam abandonados em futuro próximo. Esta é a opinião, por exemplo, de Patterson (Patterson, D. A., in *Microprocessors in 2020*, Scientific American 273, 3, 1995, p. 48-51), um dos cientistas que mais contribuíram para a concepção de modernos circuitos integrados" **[Kow96]**.

---

* A expressão usada por Backus tornou-se popular e passou a denotar, de maneira genérica, o fato de que a eficiência de processamento das máquinas com a concepção introduzida por von Neumann é limitada por problemas de comunicação entre a memória e as outras unidades. É interessante notar que no documento em que descreve o EDVAC, o próprio von Neumann utiliza a palavra *gargalo* quando comenta as dificuldades de projeto e funcionamento de memória **[Kow96]**.

7 A revolução do hardware e do software

Os primeiros computadores da década de 1940 possuíam somente dois níveis de linguagem de programação: o nível da linguagem de máquina, no qual toda a programação era feita, e o nível da lógica digital, onde os programas eram efetivamente executados. Com Maurice V. Wilkes, em 1951, na Universidade de Cambridge, surgiu a idéia de se projetar um computador de três níveis, a fim de se simplificar o hardware. Essa máquina, conhecida como EDSAC (Electronic Delay Storage Automatic Calculator), tinha um programa denominado *interpretador*, armazenado permanentemente, cuja função era executar os programas em linguagem de máquina. O hardware assim poderia ser simplificado: teria apenas que executar um pequeno conjunto de microinstruções armazenadas*, o que exigia menos circuitos eletrônicos. A partir daí começaram a evoluir as linguagens e as arquiteturas das máquinas.

7.1 Da segunda geração de grandes computadores aos dias de hoje

A segunda geração (1956 - 1963) foi impulsionada pela invenção do *transistor* (1948) e em 1956 já se produziam computadores com esta tecnologia. Apareceram também os modernos dispositivos, tais como as impressoras, as fitas magnéticas, os discos para armazenamento, etc. Os computadores passaram a ter um desenvolvimento rápido, impulsionados principalmente por dois fatores essenciais: os sistemas operacionais e as linguagens de programação.

Os circuitos integrados propiciaram um novo avanço e com eles surgiram os computadores de terceira geração (1964 - 1970). As tecnologias LSI, VLSI e ULSI† abrigam milhões de componentes eletrônicos em um pequeno espaço ou *chip*, iniciando a quarta geração, que vem até os dias de hoje.

Os atuais avanços em pesquisa e o projeto de novas tecnologias para os computadores estão possibilitando o surgimento da quinta geração. Dois avanços que configuram um divisor de águas são o *processamento paralelo*, que quebrou o paradigma de von Neumann, e a tecnologia dos supercondutores.

7.2 O desenvolvimento das linguagens

Várias linguagens, muitas delas conceitualmente diferentes entre si, foram surgindo e sendo aprimoradas, incorporando-se umas em outras. Com algumas poucas exceções, o projeto de cada linguagem foi influenciado pela experiência em linguagens anteriores. Merecem atenção especial, pelo seu pioneirismo e pelos novos paradigmas que introduziram, as linguagens alto nível FORTRAN e LISP.

---

* As instruções do programa em linguagem de máquina seriam convertidas em conjuntos dessas microinstruções, que então são executadas.

† Large Scale Integration, Very Large Scale Integration, Ultra Large Scale Integration.

Com relação ao FORTRAN, em 1954 realizou-se um simpósio sobre 'computação automática'*, e seu maior evento foi a apresentação do compilador algébrico de Laning e Zierler (ver *Os primeiros 'compiladores'*). Foi o primeiro 'software' que permitiu como entrada de dados um código algébrico elegante, embora limitado. Nesse meio tempo John Backus já montara um grupo de pesquisa dentro da IBM para trabalhar em um projeto sobre programação automática, a fim de responder a uma questão fundamental: "(...) pode uma máquina traduzir uma linguagem matemática abrangente em um conjunto razoável de instruções, a um baixo custo, e resolver totalmente uma questão?" **[Wex80]**. Em novembro de 1954 a equipe de Backus tinha criado o *IBM Mathematical FORmula TRANslation System, o FORTRAN*. O primeiro parágrafo da apresentação desse trabalho enfatizava que os sistemas anteriores ofereciam duas escolhas: ou uma fácil codificação e uma execução lenta do programa ou uma laboriosa codificação com rápida execução, mas "o FORTRAN propiciava o melhor das duas opções" **[KP80]**. Com o FORTRAN apareceram as expressões simbólicas, subprogramas com parâmetros, mas principalmente ocorreu a primeira tentativa de se definir rigorosamente a sintaxe de uma linguagem de programação. Um pouco mais tarde surgiu a notação BNF para a descrição sintática de uma linguagem de programação.

A história do LISP remonta a Turing e Church. Pela análise de Turing nos anos de 1936 e 1937, após seu famoso artigo sobre o décimo problema de Hilbert, o cálculo-lambda de Church, apesar da sua sintaxe simples, era suficientemente poderoso para descrever todas as funções mecanicamente computáveis, ou seja, pode ser visto paradigmaticamente como uma linguagem de programação. No cálculo-lambda, muitos problemas de programação, especificamente aqueles referentes às chamadas de procedimento, estão em sua forma mais pura, e isto influenciará diretamente linguagens como LISP e Algol **[Bar84]**.

Em 1955 e 1956 E.K. Blum, no U.S. Naval Ordinance Laboratory desenvolveu uma linguagem completamente diferente das demais, ADES (Automatic Digital Encoding System), baseada na teoria das funções recursivas e no esquema desenvolvido para elas por Kleene. Foi a primeira linguagem "declarativa", no sentido de que o programador estabelece as relações entre as variáveis quantitativas sem explicitamente especificar a ordem de avaliação (mais à frente se falará sobre este paradigma de programação).

Aparece agora a figura de John McCarthy, matemático, um dos primeiros a trabalhar no tema de Inteligência Artificial. Juntamente com Marvin Minsky iniciou um grande projeto nessa área. Estava procurando desenvolver uma linguagem algébrica para processamento de listas, preocupado com o problema de como representar informações da realidade por meio de sentenças escritas em uma linguagem formal adequada, e de como criar um programa que executasse fazendo inferências lógicas. Surgiu então o LISP, uma linguagem que pode ser utilizada como um formalismo para descrição de algoritmos, para escrever programas e provar propriedades de algoritmos, sendo adequada à computação simbólica e à inteligência artificial. Sobretudo com LISP pode-se visualizar melhor um importante conceito na computação moderna que é o uso de estruturas de dados como objetos abstratos. É um dos

---

* Aparece aqui novamente este termo, utilizado por Knuth **[KP80]** e John Backus **[Wex80]** e, de acordo com este último significava naqueles primeiros tempos "para muitos simplesmente escrever códigos mnemônicos e endereço simbólico, para outros o simples processo de acessar subrotinas de uma biblioteca e inserir nelas os endereços dos operandos. A maior parte dos sistemas de 'programação automática' eram programas de montagem ou conjuntos de subrotinas ou os sistemas interpretativos (...)"**[Wex80]**.

aspectos centrais dessa linguagem, comparada a como a Matemática usa os números naturais como entidades abstratas.

Nos inícios da década de 1960, fruto do trabalho de americanos e europeus, surgiu uma linguagem projetada para representar algoritmos ao invés de se escrever programas simplesmente, o *Algol-60*. Ela implementava o conceito de estrutura de blocos, onde variáveis, procedimentos, etc., poderiam ser declarados onde quer que o programa os necessitasse. Algol-60 influenciou profundamente muitas linguagens que vieram depois e evoluiu para o Algol-68.

PL/I surgiu como uma tentativa de se projetar uma linguagem de uso geral reunindo características de linguagens para aplicações numéricas como FORTRAN e Algol e para processamento de dados comerciais. Ela inovou ao permitir a construção de código de 'baixo nível' para o controle de exceções e o conceito de processamento concorrente, entre outros. O resultado foi algo anômalo, complexo e incoerente, de difícil implementação.

Foi a linguagem *Pascal* entretanto que se tornou a mais popular das linguagens do estilo Algol, por ser simples, sistemática e facilmente implementável nos diferentes computadores. O Pascal, junto com o Algol-68, está entre as primeiras linguagens com uma ampla gama de instruções para controle de fluxo, definição e construção de novos tipos de dados. *Ada*, que veio depois do Pascal, introduziu o conceito de pacotes e permite a construção de grandes programas com estrutura modular.

Podem-se discernir na história das linguagens certas tendências. A primeira foi a de perseguir altos níveis de abstração. Os rótulos simbólicos e mnemônicos das linguagens de montagem abstraem códigos de operação e endereços. Variáveis e atribuição abstraem acesso a um endereço de memória e atualização. Estruturas de dados abstraem formas de armazenamento. Estruturas de controle abstraem desvios. Procedimentos abstraem subrotinas. E assim por diante.

Outra tendência foi a proliferação dos *paradigmas*. A maioria das linguagens mencionadas até agora são *imperativas*, caracterizadas por comandos que atualizam variáveis. A estrutura das linguagens imperativas é induzida pelo hardware, com preocupação de que os dados trafeguem o mais rapidamente possível. Daí alguns de seus aspectos relevantes: seqüência de comandos, atribuição, controles (loopings), etc. É ainda o paradigma dominante.

Já as linguagens que seguem o paradigma *funcional* (também conhecidas como *declarativas*), como o LISP, tem como características a clareza e a busca de um maior poder expressivo, procurando manter a maior independência possível do *paradigma de von Neumann,* que caracteriza as linguagens imperativas*. Buscam uma transparência referencial e a não ocorrência de efeitos colaterais nas suas instruções. Em LISP não há o conceito de estado – dado por uma atribuição –, memória, seqüência de instruções, etc., procurando-se tornar mais visível o uso das funções. Nas linguagens imperativas as funções dependem de estados internos, fora de seu contexto (x := x + 'argumento'), com a produção de efeitos colaterais (alteração de valores, impressão, etc.). *LISP* foi a ancestral das linguagens funcionais que

* O paradigma ou arquitetura de von Neumann refere-se ao conceito de programa armazenado, conforme documento apresentado por Neumann em junho de 1945 sobre o EDVAC. As linguagens imperativas, preocupadas com performance de execução, têm em em conta o trânsito dos dados entre a unidade central de processamento e os dispositivos onde estão armazenadas instruções e informações .

culminaram atualmente em linguagens como *Miranda, ML e Haskell,* que tratam funções como valores de primeira classe.

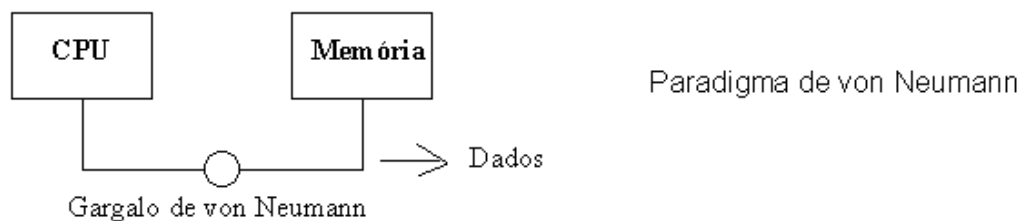



Figura 39: Gargalo de von Neumann

Smalltalk é uma linguagem baseada em classes de objetos. Um objeto é uma variável que pode ser acessada somente através de operações associadas a ele. Smalltalk é um exemplo de uma linguagem que segue o paradigma de *orientação a objeto. Simula* foi um ancestral de tais linguagens.

É importante reparar que a notação matemática em sua generalidade não é facilmente implementável. No entanto muitos projetistas de linguagens quiseram explorar subconjuntos da notação matemática em linguagens de programação. Surgiram então tentativas de se construir uma 'linguagem lógica', isto é, baseada em um subconjunto da lógica matemática. O computador é programado para inferir relacionamentos entre valores, ao invés de computar valores de saída a partir de valores de entrada. Prolog popularizou a linguagem lógica. Em sua forma pura é fraca e ineficiente, tendo sido alterada para incluir características não lógicas e tornar-se mais amigável como linguagem de programação.

No início da década de 1990 ocorreu uma difusão intensa do paradigma da *orientação a objeto*[*]. Este paradigma esteve em gestação por cerca de 30 anos e as novas tecnologias como a Internet, as necessidades geradas pela novas arquiteturas, tais como a de cliente-servidor[†] e a do processamento distribuído, coincidiam com o paradigma da orientação a objeto: encapsulamento, mensagem, etc. O crescimento da Internet e o "comércio eletrônico" introduziram novas dimensões de complexidade no processo de desenvolvimento de programas. Começaram a surgir linguagens que buscam superar esses novos desafios de desenvolvimento de aplicações em um contexto heterogêneo (arquiteturas de hardware incompatíveis, sistemas operacionais incompatíveis, plataformas operando com uma ou mais interfaces gráficas incompatíveis, etc.). Apareceram *C++* e linguagens como *Eifell, Objective C, Cedar/Mesa* (elaborada pela Xerox, para fazer pesquisa de dados), *Delphi* (uma evolução da linguagem Pascal) entre outras. E, "o próximo passo ou um paradigma completamente novo" **[GM95]**, surge a linguagem JAVA.

[*] Falando de uma maneira mais técnica e bastante genérica, significa que o foco da atenção do programador recai mais nos dados da aplicação e nos métodos para manipulá-los do que nos estritos procedimentos.

[†] Em termos gerais significa o partilhamento de uma aplicação em duas. A interface do usuário e a maioria dos programas é executada no cliente, o qual será provavelmente uma estação de trabalho ou um PC de alta performance. Os dados da aplicação residem no servidor, provavelmente em um banco de dados de um computador de grande porte. Desta maneira mantêm-se os dados onde podem ser melhor protegidos, atualizados, salvos, enquanto que o poder computacional fica distribuído diretamente pelas mesas de trabalho dos 'clientes'.

A origem da Java está ligada a um grupo de pesquisa e desenvolvimento da Sun Microsystems formado em 1990, liderado por Patrick Naughton e James Gosling, que buscava uma nova ferramenta de comunicação e programação independente da arquitetura de qualquer dispositivo eletrônico. Em 1994, após o surgimento do NCSA Mosaic e a popularização da Internet, a equipe redirecionou os seus esforços a fim de criar uma linguagem para aplicações multimídia *on line*.

Conforme Linden **[Lin96],** Java foi inspirada por várias linguagens: tem a concorrência da Mesa, tratamento de exceções como Modula-3, linking dinâmico de código novo e gerenciamento automático de memória como LISP, definição de interfaces como Objective C, e declarações ordinárias como C. Apesar dessas qualidades, todas importantes, na verdade duas outras realmente fazem a diferença e tornam Java extremamente atrativa: sua *portabilidade* e o novo conceito de *arquitetura neutra*.

Portabilidade significa que Java foi projetada objetivando aplicações para vários sistemas heterogêneos que podem compor uma rede como a Internet, por exemplo, e as diferentes características dessa rede. Java procura obter os mesmos resultados de processamento nas diferentes plataformas.

Por arquitetura neutra entende-se que programas em Java são compilados para se obter um código objeto (*byte code* na terminologia Java) que poderá ser executado em um Power PC que use o sistema operacional OS/2, ou em um sistema baseado no chip Pentium debaixo do Windows 95 ou em um Macintosh usando MacOs, ou em uma estação de trabalho Sparc rodando Unix. Ou seja, em qualquer computador, desde que tal computador implemente o ambiente necessário para isso, denominado conceitualmente de *Máquina Virtual Java*.

Com a linguagem Java se começou a superar barreira que impedia que a Internet se tornasse um computador: a barreira que impedia o uso de um software utilizado em um determinado lugar, executando-o em qualquer plataforma.

7.3 Arquiteturas de computadores e sistemas operacionais

O termo *arquitetura* de computador vem da possibilidade de se visualizar uma máquina como um conjunto hierárquico de níveis que permite entender como os computadores estão organizados. Os primeiros computadores digitais por exemplo somente possuíam dois níveis. O primeiro é chamado o nível da *lógica digital*, formado no início por válvulas e depois por transistores, circuitos integrados, etc. O segundo é chamado de nível 1, também chamado de nível de microprograma, que é o nível da linguagem da máquina, onde toda a programação era feita, através de zeros e uns, e que posteriormente seria o responsável por interpretar as instruções do nível 2.

Com Maurice Wilkes em 1951 surgiu outro nível, onde as instruções eram escritas de um modo mais conveniente para o entendimento humano: a técnica consistia em substituir cada instrução desse novo nível por um conjunto de instruções do nível anterior (nível da máquina) ou examinar uma instrução de cada vez e executar a seqüência de instruções equivalentes do nível da máquina. Denominam-se estes procedimentos por *tradução* e

interpretação. Isto simplificou o hardware que agora somente tinha um conjunto mínimo de instruções e portanto menos circuitos eram necessários.

A partir daí a evolução do hardware avança juntamente com as novas descobertas científicas: quase na mesma época do aparecimento dos transistores, por exemplo, surgiu o conceito de *barramento de dados,* que acelerou a velocidade dos computadores. Ao mesmo tempo apareceram os grandes *sistemas operacionais*, (simplificadamente, um sistema operacional é um conjunto de programas mantidos no computador durante todo o tempo, liberando o programador de tarefas relacionadas diretamente com o funcionamento da máquina), como o DOS e OS, da IBM. Estes evoluíram possibilitando novos conceitos que melhoraram a performance das máquinas, como por exemplo os *sistemas de multiprogramação*, isto é, a possibilidade de vários programas serem executados em paralelo em uma mesma da máquina. Se um destes programas tiver origem em um terminal remoto, tal sistema será chamado de *tempo compartilhado*. Um importante marco que possibilitou esses avanços foi a introdução de processadores de entrada e saída, também chamados de *canais*. Isso motivou o aparecimento dos conceitos de concorrência, comunicação e sincronização: uma vez que dois processadores estão operando simultaneamente, surge a necessidade de prover mecanismos para sincronizá-los e estabelecer um canal de comunicação entre eles.

É a era das arquiteturas *mainframes:* o suporte às tarefas computacionais e o desenvolvimento das aplicações são feitos numa área central, denominada *centro de computação*. Terminais conectados diretamente à máquina são utilizados somente por pessoas relacionadas às aplicações disponíveis.

Nos anos 70 surgiram os supercomputadores, máquinas que inovaram na arquitetura. Até o momento, o crescimento da eficiência dos computadores estava limitado pela tecnologia, mais especificamente pelo *processamento escalar* que exigia que o processador central de um computador terminasse uma tarefa para começar a realizar outra, produzindo o gargalo de von Neumann. Um avanço significativo veio com o supercomputador Cray-1, da Cray Research[*], em 1971. Foi a primeira *máquina pipeline*, cujo processador executava uma instrução dividindo-a em partes, como na linha de montagem de um carro. Enquanto a segunda parte de uma instrução estava sendo processada, a primeira parte de outra instrução começava a ser trabalhada. A evolução seguinte foi a denominada *máquina vetorial*, ou *máquina SIMD* (single instruction multiple data) cujo processador trabalhava com mais de um conjunto de dados ao mesmo tempo. Um pouco depois surgiu a arquitetura *MIMD* (multiple instructions multiple data) e apareceram máquinas com múltiplos processadores como a Connection Machine, com 65.536 processadores[†].

[*] Muito da história dos primeiros tempos dos supercomputadores coincide com a história daquele que é considerado o pai dos supercomputadores, Seymour Cray (1926-1996), fundador da Cray Research, que liderou a construção dos computadores mais rápidos do mundo durante vários anos. Seymour Cray inventou ou contribuiu diretamente na criação de múltiplas tecnologias usadas pela indústria dos supercomputadores, entre as quais está: a tecnologia de vetor de registradores no CRAY-1, a tecnologia do semicondutor de gálio arsênico e a arquitetura RISC (Reduced Instruction Set Computing).

[†] Deve-se observar que apesar da capacidade de execução paralela de centenas de tarefas, dependendo de como é feita a comunicação entre os processadores, a eficiência de tais máquinas pode ser frustrante e as pesquisas continuam em busca do aumento dessa eficiência.

Há primariamente três limites para a performance dos supercomputadores: a velocidade do processador, o gasto de tempo (o termo técnico, amplamente utilizado na Computação, é *overhead*), que envolve fazer um grande número de processadores trabalharem juntos em uma única tarefa, e a velocidade de entrada e saída entre os processadores e entre os processadores e a memória*. A velocidade dos processadores aumenta a cada dia, mas a um alto custo de pesquisa e desenvolvimento, e a realidade é que se está alcançando os limites dos processadores baseados em silício. Seymour Cray demonstrou que a tecnologia de gálio arsênico poderia ser a solução, mas é muito difícil trabalhar com ele e poucas indústrias estariam aptas a desenvolver processadores desse tipo. A solução, como se falará mais adiante caminha para o uso de um maior número de processadores, dando maior velocidade ao computador pelo emprego do *processamento paralelo*.

Com a tecnologia VLSI (Very Large Scale Integration, quarta geração de computadores) surgiram os minicomputadores, o que possibilitou muitas empresas e universidades informatizarem seus departamentos. Os grandes usuários interligavam os minicomputadores para enviar tarefas aos seus mainframes. A arquitetura principal continuava no entanto estabelecida no centro de computação. Do minicomputador para o computador pessoal foi somente um passo, e no início da década de 1980 apareceram os primeiros PC's. Ainda nos anos de 1980 apareceram as arquiteturas RISC (Reduced Instruction Set Code), com a promessa de ganho de desempenho pela eliminação do conceito de microprograma. De qualquer maneira essas máquinas ainda são máquinas de von Neumann tradicionais, com todas as suas limitações, a maior delas a velocidade dos circuitos que não pode crescer indefinidamente.

As tentativas de quebrar o gargalo de von Neumann e o início da descentralização dos sistemas, com o surgimento das arquiteturas de rede que possibilitaram a universalização do uso da tecnologia da Computação, fizeram emergir e desenvolver as arquiteturas paralelas de hardware.

A idéia de incluir paralelismo nos computadores é tão antiga quanto os próprios computadores. Trabalhos desenvolvidos por von Neumann na década de 1940 já discutiam a possibilidade de algoritmos paralelos para a solução de equações diferenciais. O sistema Model V, desenvolvido entre 1944 e 1947 por G. R. Stibitz e S. B. Willians nos laboratórios da Bell Telephone é um exemplo típico de máquina paralela. Constituído por dois processadores e com três posições de entrada e saída, esse multiprocessador primitivo tanto era capaz de executar dois programas distintos quanto era possível que os dois processadores ficassem alocados para um mesmo programa. Posteriormente foi desenvolvido o Illiac IV, na década de 1960, constituído por 64 processadores. Como foi citado, a partir da década de 1970 começaram a ser produzidos supercomputadores baseados em arquiteturas paralelas.

Juntamente com as arquiteturas evoluíram os sistemas operacionais e a evolução das linhas de processadores de uma empresa como a Intel servem para refletir a evolução da indústria dos computadores em um determinado período. Como destaque podem-se citar o

---

* A velocidade de entrada/saída entre a memória principal (tecnicamente conhecida como RAM) e os dispositivos de armazenamento é um problema que afeta todos os tipos de computadores. Mas como os supercomputadores tem uma grande quantidade de memória principal, esse problema pode ser resolvido facilmente com um gasto mais generoso de dinheiro.

MS-DOS, o OS/2 e o UNIX. Especialmente este último, que surgiu como fruto dos trabalhos de um engenheiro da Bell Labs, Ken Thompson, foi popularizado nos meios universitários que usavam computadores PDP-11/45, durante a década de 1970. A palavra UNIX espalhou-se rapidamente por todo o mundo e no início de 1980 este sistema operacional estava disponível em mais máquinas do que qualquer outro sistema operacional da época, continuando hoje ainda a ser amplamente utilizado.

A mais recente evolução da Computação foi o resultado da rápida convergência das tecnologias de comunicação de dados, de telecomunicação e da própria informática. É a Internet, ou o modelo computacional baseado em uma rede, que teve suas origens nos anos da década de 1970, como um esforço do Departamento de Defesa dos EUA para conectar a sua rede experimental, chamada ARPAnet, a várias outras redes de rádio e satélites. Espalhou-se logo em seguida nos meios acadêmicos e está bastante popularizada.

7.4 Uma nova mentalidade

A partir de 1975, com a disseminação dos circuitos integrados, a Computação deu um novo salto em sua história, proporcionado pelo surgimento e desenvolvimento da indústria dos computadores pessoais e, principalmente, pelo aparecimento da computação multimídia. Com o aparecimento dos microcomputadores, rompeu-se a barreira de deslumbramento que cercava as grandes máquinas e seu seleto pessoal que as manipulava, e surgiu a possibilidade da transferência do controle do computador para milhares de pessoas, assistindo-se à sua transformação em um bem de consumo.

Em 1975 a revista americana *Popular Eletronics* anunciou a chegada do □Primeiro kit de minicomputador do mundo a rivalizar com os modelos comerciais□. Tratava-se do Altair 8800, construído com base no chip 8080 da Intel por H. Edwards Roberts, oficial da Força Aérea americana, graduado em engenharia elétrica. O sucesso foi imediato: 4.000 unidade vendidas em três meses.

Impulsionados pelo sucesso, um jovem programador, Paul Allen, associa-se a um estudante de Harvard, Willians Gates, com o objetivo de escrever uma versão popular de uma linguagem computacional, o Basic, para o Altair. Mais tarde ambos fundaram a Microsoft, que se tornou na década de 1990 a mais bem sucedida empresa de software da história dos microcomputadores. Outro grande sucesso dos primeiros anos dos computadores pessoais foram os microcomputadores lançados pela Apple, nascida em 1976, fundada pela dupla Steve Jobs e Stephen Wozniac, que foi um sucesso total naqueles primeiros anos.

Mas realmente essa nova □onda□ só foi possível graças à entrada da IBM na competição, quando, em 12 de agosto de 1981, em Nova Iorque, executivos da □big blue□, como é conhecida, apresentaram o novo computador do momento, o IBM PC (Personal Computer). Talvez o fato mais importante, que afetaria pelos próximos anos o panorama da indústria dos microcomputadores foi a decisão da IBM de utilizar uma □arquitetura aberta□: selecionar os componentes básicos e o sistema operacional de fontes externas à IBM.

Contando um pouco dessa história. A força-tarefa que a IBM tinha designado para a criação do computador pessoal decidiu que queria um computador de 16 bits, mais potente e mais fácil de programar que as máquinas de oito bits então existentes. A Intel havia anunciado, então recentemente, o chip 8086 de 16 bits, mas a IBM, temendo que fizesse

sombra aos demais itens já comercializados por ela, escolheu o 8088, uma versão do chip com barramento de 8 bits e estrutura interna de 16 bits. Tal tecnologia proporcionava ainda a vantagem de trabalhar com as placas de expansão de oito bits existentes no mercado e com dispositivos de oito bits relativamente baratos, como os chips controladores. Na busca pelo software a IBM foi às portas da Digital Research para ver a possibilidade de portar seu sistema operacional – de grande sucesso! – CP/M para a arquitetura 8086, mas esta rejeitou o contrato de exclusividade apresentado pela IBM. Assim, a equipe da IBM rumou para os escritórios da Microsoft, de quem esperavam obter uma versão do BASIC e acabaram assinando um contrato não só deste software mas também sobre o sistema operacional. A Microsoft adquiriu e incrementou um sistema operacional 8086 da Seattle Computer Products – o QDOS – licenciando-o para a IBM, que começou a comercializá-lo com o nome de PC-DOS.

Os anos da década de 1980 poderiam ser caracterizados pelo aperfeiçoamento de softwares – tanto sistemas operacionais como utilitários: planilhas, editores de texto, e outros mais – para o padrão DOS e o desenvolvimento de um mercado de "clones" de diferentes tipos de máquinas que seriam capazes de executar os programas elaborados para o padrão. A Apple continuava a fazer sucesso com sua família Apple II, embora fracassando na introdução do Apple III e do formidável LISA, a primeira tentativa de popularizar a combinação de mouse, janela, ícones e interface gráfica com usuário. Mas o preço de US$10.000,00 assustou e espantou o mercado.

O próximo passo a ser dado – sem contar a evolução e aprimoramento do hardware sem o qual isso não seria possível – seria a gradual passagem dos aplicativos para ambiente DOS – verdadeiro mar de produtos – para um novo padrão de ambiente, que começava a ganhar contornos definitivos, e que protagonizou o início de uma nova idade na história dos microcomputadores: o do sistema operacional Windows, que tornou-se padrão dominante para os aplicativos para PC, tornando a Microsoft líder na definição de especificações multimídia. É importante no entanto fazer-se justiça: o padrão Windows inspirou-se no padrão Macintosh, lançado pela Apple em 1984: um computador que era capaz de oferecer mais de um "prompt" de DOS e uma interface baseada em caracteres; ele podia ter várias janelas, menus suspensos e um mouse. Infelizmente o Macintosh não era compatível com os programas e aplicativos já existentes e não era expansível.

7.5 A Computação como Ciência

Ao lado dessa evolução do hardware e do software, a Computação abriu-se em leque e novas tendências surgiram dentro dela, incorporando estas duas entidades.

A Inteligência Artificial, a Teoria da Complexidade Computacional[*] e a Teoria de Bancos de Dados abriram novos campos de estudo. Na década de 1960 a Ciência da

[*] A Teoria da Complexidade Computacional é um ramo da Computação que estuda o grau de dificuldade envolvido na resolução algorítmica de classes de problemas. Um dos principais tópicos abordados diz respeito à eficiência (em termos de tempo) envolvida na execução de um algoritmo.

Computação tornou-se uma disciplina verdadeira. A primeira pessoa a receber um título de Ph. D. de um departamento de Ciência da Computação, foi Richard Wexelblat, na Universidade da Pensilvânia , em 1965. Consolidaram-se os estudos sobre a Teoria dos Autômatos e a Teoria de Linguagens Formais, principalmente com Noam Chomsky e Michael Rabin. O nascimento do ramo das especificações formais, que introduziu um novo paradigma no desenvolvimento de sistemas computacionais, veio dentro dessa década, com o início das buscas pela corretude de programas através do uso de métodos formais.

R. W Floyd, em 1967, propôs que a semântica de linguagens de programação fosse definida independentemente dos processadores específicos a que se destina aquela linguagem. A definição pode ser dada, segundo Floyd, em termos do método para a prova de programas expresso na linguagem. O seu trabalho introduziu o que passou a ser conhecido como o método das anotações (assertivas) indutivas para a verificação (prova) de programas e uma técnica envolvendo "conjuntos com ordenação bem fundada para provar o término de um programa"*.

Uma extensão das idéias de Floyd foi proposta por C. A. Hoare em 1969. Hoare formulou uma teoria axiomática de programas que permite a aplicação do método das invariantes de Floyd a textos de programas expressos em linguagens de programação cuja semântica é precisamente formulada. Este trabalho tornou-se ainda um dos fundamentos do que se chamou mais tarde "programação estruturada". Dijkstra desenvolveu a idéia de que a definição (no estilo proposto por Hoare) pode ser usada para a derivação (síntese) de um programa e não apenas para sua verificação **[Luc82]**.

A partir dessas pesquisas surgiu a Engenharia de Software, que busca garantir a corretude na construção de sistemas. O desenvolvimento de sistemas computacionais até então era feito de uma maneira quase que artesanal. Não havia critério orientativo algum durante o processo. Isso acabou sendo fatal, como o revelaram certos estudos, elaborados na década de 1970, sobre o desenvolvimento de sistemas: ausência de corretude e consistência, baixa qualidade, manutenção extremamente custosa em função de problemas não detectados por ausência de uma validação de requisitos mais rigorosa, não reaproveitamento de código, prazos de implementação não cumpridos em conseqüência de erros detectados ao longo dessa mesma fase de implementação, etc.

Obedecendo a um grau de formalização maior, apareceram como primeira reação a essa abordagem *informal*† modelos e métodos de desenvolvimento de sistemas chamados estruturados, que na verdade são conjuntos de normas e regras que guiam as várias fases de desenvolvimento de sistemas e as transições entre elas. É a abordagem *sistemática*. Ainda aqui não está presente um formalismo definido com regras precisas. A prototipação e a orientação a objeto são abordagens que podem ser consideradas sistemáticas.

A abordagem *rigorosa* já apresenta um sistema lingüístico formal para documentar as etapas de desenvolvimento e regras estritas para a passagem de uma etapa a outra. Não se exige que as demonstrações de corretude das transformações realizadas sejam feitas

---

* O objetivo é escolher essas proposições de tal forma que elas sejam satisfeitas cada vez que o fluxo de controle do programa passe pelo ponto anotado e de maneira que cada ciclo do fluxograma seja "cortado" (anotado) por uma proposição.

† Para estabelecer uma distinção entre as várias espécies de abordagem vamos seguir uma classificação sugerida por Bjorner **[Tan92]**, de acordo com o grau de formalização.

formalmente, bastando uma argumentação intuitiva*. E finalmente a abordagem puramente *formal*, rigorosa, com a exigência de que todas as demonstrações necessárias para garantir a corretude do processo sejam realizadas formalmente.

É necessário notar que essas duas últimas abordagens exigem um conhecimento mais profundo do raciocínio lógico formal e de um sistema lingüístico formal adequado. Embora a abordagem formal se apresente como único meio de se dar uma garantia real à atividade de construção de sistemas, muitos autores mostram-se céticos quanto ao verdadeiro impacto que ela venha a ter na prática, devido à dificuldade de aprendizado do necessário arcabouço matemático.

Figura 40: Donald E. Knuth

Donald E. Knuth iniciou nos fins dessa década um rigoroso tratado sobre as bases matemáticas para a análise de algoritmos, produzindo os três conhecidos volumes do *The Art of Computer Programming* **[Knu69]**, que propiciaram a base para o amadurecimento dos estudos da complexidade de algoritmos. Pode-se dizer que o trabalho de Knuth é um dos grandes marcos da Computação no século XX: antes de Knuth não havia um corpo sistemático do estudo da programação e dos algoritmos.

Ainda no campo da Complexidade Computacional novos avanços se deram a partir de 1971, com o trabalho de Steve Cook e Richard Karp sobre problemas NP-completos† e os estudos sobre criptografia de Ronald Rivest, Adi Shamir e Leonard Adleman. Em 1977 H.J. Bremermann desenvolveu alguns trabalhos pioneiros dentro da teoria da complexidade, mostrando os limites físicos na arquitetura de computadores de qualquer tipo e que estes

* É importante notar que a prova formal pode ser feita.

† Um problema dito P (de *polinomial*) é executado em um computador com um número de passos dado pela fórmula An^k (A e k são inteiros fixos e n é o número de dados de entrada). Algoritmos NP (de *tempo não determinístico polinomial*) executam em tempo exponencial, em um número de passos 2^n ou n! (problema do caixeiro viajante por exemplo, a solução de alguns teoremas lógicos de primeira ordem, o problema da torre de Hanói, etc.).

limites físicos atuam como fatores restritivos para a computação de determinados problemas. De acordo com ele, existe um tempo chamado *limite fundamental* para a velocidade dos computadores que não pode ser ultrapassado. Tal limite deriva-se da idéia de que a velocidade máxima de transmissão de sinal entre os componentes internos da máquina é limitada pela velocidade da luz. Mesmo que se pudessem construir máquinas muito pequenas, otimizando-se a trajetória de transmissão de sinais, esse limite não pode ser ultrapassado. E ainda que se chegue a uma máquina cuja velocidade de transmissão seja próxima à da luz, existem problemas computacionais que são intratáveis, como por exemplo os "problemas NP": mesmo com a velocidade da luz tais problemas poderiam levar a idade do universo para serem processados **[Tei97]**.

7.6 A inteligência artificial

É um dos ramos da Ciência da Computação merecedor de especial destaque, pela sua gama de influência nas pesquisas e novas áreas que se abriram a partir do seu início. É aquela área da Computação, em termos mais gerais, voltada para o estudo de técnicas de construção de programas que permitam ao computador simular aspectos do comportamento da inteligência, tais como jogar xadrez, provar teoremas lógicos, compreender partes específicas de uma linguagem natural como, por exemplo, o português, etc.

Os primórdios da Inteligência Artificial (IA a partir de agora) remontam à década de 1940. Começou a predominar nesses anos o movimento ciberneticista, que acreditava entre outras coisas que a atividade humana poderia um dia ser estudada por meio de modelos matemáticos, como se faz com outros tantos fenômenos da natureza. Seguindo os trabalhos de Gödel, muitos matemáticos imbuíram-se do objetivo de formalizar a noção de procedimento e definir o que poderia ser feito através de um algoritmo, e vieram a tona os trabalhos de Turing, Church, Kleene e Post*. Os resultados desses esforços acabaram por ser equivalentes e se estabeleceram os limites do que é computável†.

No ano de 1943 foram publicados os trabalhos de Warren McCulloch e Walter Pitts, que propuseram um modelo de neurônios artificiais, onde cada neurônio era caracterizado como sendo "on" ou "off", e era representado por uma chave onde "on" seria a resposta a estímulos por um dado número de neurônios vizinhos. Eles mostraram que qualquer função computável poderia ser processada por algum tipo de rede de neurônios conectados e que os conectivos lógicos poderiam ser implementados através de estruturas de rede simples **[MP43]**. Estão presentes aqui as pesquisas de Claude Shannon, que entre outras coisas descreveu em termos lógicos o funcionamento de certos sistemas físicos, e vice-versa: sistemas físicos que poderiam representar um raciocínio lógico. A fusão das idéias de Shannon e Boole, associadas a um tratamento simplificado do neurônio do cérebro humano,

---

* Como foi visto, Turing desenvolveu a Máquina de Turing, Church desenvolveu o cálculo-lambda (que forneceu a base para a linguagem LISP, desenvolvida por McCarthy, uma das favoritas do pessoal da IA), Kleene desenvolveu a teoria das funções recursivas, enquanto Emil Post introduziu sistemas para reescrita de cadeias de símbolos (a gramática de Chomsky é um caso particular disso).

† Como um dos subprodutos do trabalho de Church, ficou estabelecido que tudo aquilo que um ser humano possa fazer manipulando símbolos, seguindo um finito e bem definido conjunto de regras, uma máquina equipada com o conveniente programa também poderá fazê-lo.

tornou possível o trabalho de McCulloch e Pitts, que propuseram um modelo de neurônio artificial (há um trabalho sobre este assunto em **[Arb87]**). Queriam esses dois pesquisadores mostrar que se os neurônios artificiais pudessem efetuar computações lógicas, estaria aberto o caminho para simular o raciocínio humano.

Os trabalhos de Warren McCulloch e Walter Pitts tiveram grande sucesso e outros trabalhos apareceram, mas logo foram objeto de fortes críticas. Frank Rosemblatt e seus colegas da Universidade de Cornell projetaram uma máquina parecida com o modelo de MccUlloch e Pitts, denominada *Perceptron*[*]. Marvin Minsky e Seymour Papert mostraram algumas limitações dos perceptrons, como por exemplo a impossibilidade de executarem operações lógicas do tipo “ou exclusivo”. De qualquer maneira surgiu uma primeira vertente da emergente IA: a que buscava a simulação do cérebro humano do ponto de vista físico, para simular a atividade mental, e que fará surgir anos mais tarde, na década de 1970, a Ciência Cognitiva ou Conexionismo, que está apoiada em um paradigma da IA de processamento serial da informação e no “approach” da manipulação simbólica para a lingüística.

Outra vertente por onde se encaminharam os estudos da IA foi a chamada Inteligência Artificial *simbólica*, que buscava a representação e a simulação dos estados e do comportamento mentais através de programas computacionais. A representação e simulação da inteligência não estaria na construção de um determinado tipo de hardware, mas no desenvolvimento de um software que opere sobre dados. Isto teve profundos reflexos nas pesquisas posteriores e suscitou inúmeros debates, sobretudo na filosofia, sobre as relações entre a mente e o cérebro[†]. Em **[Tei97]**, há um capítulo sobre as grandes objeções levantadas ao termo IA no *sentido forte*, o qual diz que um computador adequadamente programado é uma mente e reproduz estados mentais.

Em 1950 Turing introduziu através de um artigo, *Computing Machinery and Intelligence*, o chamado ‘Turing Test”[‡], considerado também um dos primeiros esforços no campo da Inteligência Artificial **[RN95]**. Mais tarde o próprio Turing (1953) escreveu um programa de xadrez para computadores que tinham a arquitetura de von Neumann. Ao mesmo tempo, Marvin Minsky e Deam Edmonds, do Departamento de Matemática de Princeton, construíram o primeiro computador baseado em rede neural, em 1951, o SNARC. Era um computador analógico para simular uma rede de 40 neurônios **[RN95]**.

John McCarthy*,* outra figura influente da IA, após formar-se em Princeton, dirigiu-se ao Dartmouth College e convenceu Minsky, Shannon e Nathaniel Rochester, um pesquisador da IBM, a ajudá-lo a trazer pesquisadores interessados em teoria dos autômatos, redes neurais e no estudo da inteligência. Allen Newell e Herbert Simon, da Carnegie Mellon

---

[*] Pode-se imaginar um Perceptron como um dispositivo para o reconhecimento de um conjunto de padrões, não específico, isto é, com capacidade de “aprender” a reconhecer os padrões de um conjunto após um número finito de tentativas.

[†] Na verdade tais indagações remontam ao filósofo Renè Descartes que introduziu a primeira fissura no pensamento filosófico de até então, ao cavar um fosso profundo entre a matéria e o espírito humano.

[‡] Ele propôs uma definição de “pensamento” usando um jogo: um homem teria de decidir, baseado em uma conversa via teletipo, se a entidade que estava na sala ao lado, respondendo a um teste, era um ser humano ou um computador. Se a distinção não pudesse ser feita, então poderia ser dito que o computador estava “pensando” **[Tur36]**.

University elaboraram o LT (Logic Theorist)*, um programa capaz de trabalhar não-numericamente e que provou a maioria dos teoremas do segundo capítulo do Principia Mathematica, de Russell e Whitehead. Em 1952, Arthur Samuel mostrou que os computadores não fazem somente o que se lhes pede, mas são capazes de "aprender". Outros programas provadores de teoremas se seguiram ao LT e em 1958, com McCarthy surgiu o **LISP,** que se tornou a linguagem de programação predominante para IA a partir daí.

McCarthy seguiu para Stanford na busca da representação do raciocínio através da lógica formal (seu trabalho recebeu grande impulso quando J. A. Robinson elaborou um algoritmo completo para a prova de teorema na lógica de primeira ordem). Estas aplicações da lógica incluíam robótica e um projeto para demonstrar a completa integração entre raciocínio lógico e atividade física. Minsky estava mais interessado em descobrir programas para resolver problemas que exigiam inteligência para serem solucionados: problemas de analogia geométrica que aparecem nos testes de QI, problemas de álgebra, etc.

Os anos de 1966 a 1974 foram marcados por um certo ceticismo diante das dificuldades que começaram a ser encontradas, como por exemplo a não tratabilidade de muitos problemas de IA, acentuada pelos primeiros estudos dos problemas não-polinomiais determinísticos, NP, e pelas limitações das estruturas básicas usadas para gerar comportamento inteligente, como por exemplo os algoritmos de aprendizado por *back-propagation* **[RN95]**.

A década de 1970 marcou a busca pelos *sistemas baseados em conhecimento* e pelos *sistemas especialistas*, protagonizada inicialmente por Ed Feigenbaum, Bruce Buchanan e Joshua Lederberg. Os sistemas especialistas são solucionadores de problemas acoplados a grandes bancos de dados onde o conhecimento humano específico sobre determinado assunto encontra-se armazenado. O sistema deverá fornecer respostas a consultas, dar conselhos a um leigo sobre um determinado assunto, auxiliar especialistas, ensinar, etc. A idéia subjacente é que a inteligência não é somente raciocínio mas também memória. A grande meta é a preservação do conhecimento de especialistas após a morte destes. O problema, ainda em aberto, é a difícil tarefa de se representar o conhecimento, aliás nome de uma nova área surgida dentro da IA para solucionar os inúmeros problemas surgidos, principalmente os de como representar o "senso comum", o "sexto sentido" ou ainda a intuição, termos que resistem a uma conceituação clara. Ou ainda qualquer tipo de conhecimento não representável por uma expressão simbólica como ensinar alguém a jogar bola. Como formalizar estas coisas? Mais ainda: até que ponto a formalização é um instrumento eficiente para a representação do conhecimento? De qualquer maneira surgiram os sistemas especialistas para diagnóstico médico, manipulação de linguagens, etc.

Lembrando o trabalho de H.J. Bremermann citado no item anterior (*A Computação como Ciência*), sobre os limites físicos que impedem a construção de um dispositivo com velocidade 'ilimitada' (maior que a da luz por exemplo), deve-se reparar que esses mesmos limites estão presentes também dentro das reações químicas e nos impulsos elétricos que se dão nas complexas conexões dos neurônios do cérebro. Se a mente humana consegue resolver determinados problemas intratáveis (o problema da parada na máquina de Turing

---

* Newell e Simon também inventaram a linguagem IPL, para processamento de listas, para escrever o LT. Como não tinham compilador, traduziram manualmente para o correspondente código de máquina.

por exemplo) e funciona de maneira algorítmica (como pensa determinada corrente de estudiosos da simulação da mente por computador), as operações mentais tem algo a mais do que as características físicas do cérebro humano. Ou então há processamentos mentais não algorítmicos, e se cai no problema da impossibilidade de uma representação formal disso. São debates em aberto e que geram um saudável intercâmbio de idéias entre a Computação e outras áreas do conhecimento humano como a Psicologia, Biologia e Filosofia.

Também a robótica demandou estudos da área de IA, principalmente no que se refere à Visão Computacional. Os anos da década de 1980 foram os da introdução da IA na indústria e um retorno ao uso de redes neurais. De 1987 para cá houve uma mudança tanto no conteúdo quanto na metodologia das pesquisas em IA. Agora é mais comum trabalhar em teorias existentes do que a proposição de novas, e basear-se em teoremas rigorosos ou em fortes evidências experimentais do que sobre intuição. O campo do *reconhecimento da voz* é um exemplo disto.

7.7 Uma nova disciplina: a cibernética

O nascimento da cibernética como ciência está associado aos trabalhos de Norbert Wiener (1894-1964). Na II Guerra Mundial ele foi encarregado pelo governo norte-americano de resolver os problemas de controle automático da direção do tiro, na artilharia antiaérea. Wiener desenvolveu suas investigações no MIT e idealizou um sistema mediante o qual o erro, ou diferença entre o objetivo que se pretendia alcançar e o efeito real alcançado, era medido e utilizado para regular o próprio sistema, corrigindo as variáveis de velocidade, ângulo de tiro, etc. Ao procurar resolver o problema, e encontrar algumas soluções, Wiener tentou construir uma conceituação geral das questões com que lidava. Em contato com outros cientistas, como o fisiólogo Cannon, pode comprovar que, além da parte automática, esse tipo de problemas de controle e comunicação se aplicavam a muitos outros âmbitos. Por isso ampliou paulatinamente sua teoria de forma que abrangesse os seres vivos e as máquinas, ou, mais em geral, a todo corpo com dinâmica organizada pela informação.

Wiener alcunhou o termo "cibernético" tomando diretamente o vocábulo grego "kybernetiké", que significa "a arte de governar um barco". Este termo tinha um amplo uso no pensamento grego, em referência precisamente a fenômenos muito similares ao estudado pelo engenheiro norte-americano. Assim ele próprio define a cibernética como "controle e comunicação no animal e na máquina", como aparece no título de seu livro *Cybernetics* **[Wie70]**.

A Cibernética é uma teoria formada pelos aspectos relevantes de outras quatro teorias: da informação, dos jogos, do controle e do algoritmo. Estão diretamente relacionados com essa teoria os trabalhos de:

- Alan M. Turing em seus estudos sobre a possibilidade lógica das máquinas;
- Claude E. Shannon na Teoria da Informação;
- Ludwig Von Bertalanffy, biólogo que, em resultado de 30 anos de trabalhos, publicou a famosa obra intitulada "Teoria Geral dos Sistemas";
- James Watt (1736-1819), inventor do regulador centrífugo de pressão nas máquinas a vapor, germe da automatização via feedback negativo;

- John von Neumann com sua Teoria Matemática dos Jogos.

Com a cibernética surge uma disciplina que estuda a evolução temporal dinâmica dos sistemas com capacidade de auto-regulação e auto-manutenção ao interagir com o meio que o circunda. De maneira breve pode-se afirmar que as contribuições de Wiener podem resumir-se em dois pontos **[Ara78]**:

- Sublinhou a importância dos estudos interdisciplinares, mostrando o grande interesse que apresentam para cada uma das disciplinas consideradas
- Percebeu a presença de processos realimentados de controle em uma ampla classe de sistemas, tanto naturais como sociais.

Embora a cibernética como ciência não tenha como objetivo o computador – para ela é apenas mais uma das muitas estruturas existentes no universo –, ela criou, juntamente com a teoria da informação de Shannon, um novo caminho para tentar entender o homem e as máquinas. Ao se ocupar das estruturas e funções lógico-matemáticas de auto-regulação, independentemente de que estejam inscritas e se cumpram em um organismo vivo, ou em uma população humana ou em um computador eletrônico, acabou tomando parte indiretamente no desenvolvimento do hardware e do software.

A idéia de informação como uma das características fundamentais do universo levou Wiener e Shannon, separadamente, a demonstrarem que muitas coisas, desde o movimento aleatório de partículas subatômicas até o comportamento de redes baseadas em chaveamentos elétricos ou alguns aspectos do discurso humano, estão relacionados de tal modo, que podem ser expressos através de algumas equações matemáticas básicas[*].

Estas equações foram úteis na construção de computadores e redes telefônicas: muitos conceitos elaborados e delineados pela cibernética e teoria da informação tornaram-se centrais no projeto lógico de máquinas e na criação do software.

A cibernética não conseguiu estabelecer-se com um objeto e método unificados na tradição acadêmica, e o termo se utilizou cada vez menos. Seus acahados foram integrados dentro da Teoria Geral dos Sistemas[†], no que se refere aos aspectos mais teóricos. Seu lado mais prático e utilitário foram assimilados dentro da robótica. "Somente em países europeus constituiu-se como uma ciência amplíssima que engloba aspectos tão diversos como a teoria da informação, comunica'~ao, computadores, sistemas de controle, robótica, modelagem econômica, sociologia, etc. Independente da evolução acadêmica que tenha a cibernética como disciplina, é necessário referir-se a uma série de conceitos que com ela se puseram em andamento e são de uso comum em muitos âmbitos"[‡] **[Tir2002]**.

---

[*] Sobre estas idéias tão sumariamente enunciadas ver o livro Cibernética, capítulo introdutório **[Wie70]**.

[†] A Teoria Geral dos Sistemas é um completo paradigma, uma forma de pensar muito fecunda para entender a complexidade que engloba tanto os campos já citados acima que se relacionam com a Cibernética e ainda: Teoria dos Conjuntos (Mesarovic), Teoria das Redes (Rapoport), Dinâmica de Sistemas (Forrester), cfr. Bertalanffy, L. von Teoria Geral dos Sistemas **[Tir2002]**

[‡] As noções de *feed-back negativa* ou *realimentação*, *homostasis* (permanecer igual a si mesmo), *feed-before* (comportamento predictivo e por estratégia), etc.

8 A disseminação da cultura informática

Quando a História olhar para trás e estudar os anos do século XX, entre outras coisas, perceberá que, do ponto de vista científico, eles estão caracterizados como tempos em que se produziu uma aceleração tecnológica e um avanço nas comunicações sem precedentes. Não é fácil encontrar situações históricas análogas à expansão tecnológica que se assistiu nestes últimos cinqüenta anos do século. Após as revoluções do ferro, da eletricidade, do petróleo, da química, veio a revolução apoiada na eletrônica e no desenvolvimento dos computadores. A partir dos anos setenta iniciou-se a integração em grande escala da televisão, telecomunicação e informática, em um processo que tende a configurar redes informativas integradas, com uma *matriz de comunicação* baseada na informação digital, com grande capacidade de veicular dados, fotos, gráficos, palavras, sons, imagens, difundidos em vários meios impressos e audiovisuais. Pode-se até dizer que, em certo sentido, as mídias estão sendo suprimidas, pois tudo está se tornando eletrônico.

A integração dos meios de comunicação gera também uma progressiva fusão das atividades intelectuais e industriais do campo da informação. Jornalistas das redações dos grandes jornais e agências de informação, artistas, comunidade estudantil, pesquisadores trabalham diante de uma tela de computador. Em algumas sociedades, como a norte-americana por exemplo, quase 50% (dados de 1955) da população economicamente ativa está dedicada a atividades industriais, comerciais, culturais, sociais e informacionais relacionadas com coleta, tratamento e disseminação da informação. Há um aumento da eficiência informacional a cada dia, e se barateiam cada vez mais os custos tecnológicos. Não esquecendo que o computador, diferentemente das outras máquinas (que manipulam, transformam ou transportam matéria e energia) manipula, transforma e transporta um elemento muito mais limpo e menos consumidor de energia e matéria prima. Abre-se portanto uma porta para um crescimento da informação praticamente ilimitado.

Já que se está tratando principalmente neste livro sobre a evolução das idéias e conceitos que levaram ao surgimento e desenvolvimento da Ciência da Computação, pode-se falar agora de um supra-conceito maior, conseqüência que a Computação ajudou a catalisar: o surgimento da Sociedade da Informação. Sem querer adentrar no tema, merecedor de um trabalho exclusivo e com implicações históricas, antropológicas, sociológicas e até psicológicas que fogem ao presente escopo, duas considerações serão feitas: o problema do excesso de informação e o perigo do empobrecimento que pode ser causado pelo uso indevido do computador.

8.1 O domínio e o controle das informações

Existe no mundo da pintura uma expressão que se refere ao acúmulo de cores que acaba por não permitir uma clara distinção do objeto: infopoluição. Esta possibilidade começa a fazer-se realidade no âmbito da Sociedade da Informação. A informação está expandida no mundo de hoje, resultado da explosão de fontes que incluem as agências comerciais de notícias, os sistemas comerciais de satélites transmissores de imagens, a World

Wide Web, etc. Há um otimismo que leva muitos a se alegrarem com essa invasão vertiginosa de palavras, imagens e símbolos, dos quais muito poucos são controlados, de um ponto de vista semântico. Faz tempo que soam os alarmes: "Os usos normais da fala e a escrita nas sociedades ocidentais modernas estão fatalmente enfermos. O discurso que se faz nas instituições sociais, o dos códigos legais, o debate político, a argumentação filosófica e a elaboração literária, o leviatã retórico dos meios de comunicação: todos estes discursos são clichês sem vida, jargão sem sentido, falsidades intencionadas ou inconscientes. O contágio se estendeu aos centros nervosos do falar privado. Em uma infecciosa dialética de reciprocidade, as patologias da linguagem pública, especialmente as do jornalismo, a ficção, a retórica parlamentar e as relações internacionais, debilitam e adulteram cada vez mais as tentativas de psique particular dse comunicar a verdade" **[Ste91]**.

A superabundância de informação tende a mudar a natureza de cada mensagem concreta. Às vezes, a maneira mais prática de não informar é dar uma enxurrada de informações. Pode-se chegar até a privar de significado, ou tornar insignificante, a própria mensagem. A informação converte-se, nessa perspectiva, em simples ruído de fundo. Isto já ocorre, especialmente com os informes publicitários.

É uma acumulação de dados não só pela densidade de informações bem como pela sucessão rápida com que chega. Se no passado o problema era o de acesso e coleta, agora está sendo o da seleção e avaliação. A possibilidade de recolher, processar, difundir e recuperar informação de maneira quase instantânea implica numa certa desvalorização da notícia. A informação em doses exageradas acaba por tornar-se ruído. Por muita informática que exista, se não se tem capacidade de tratamento que a converta em significativa é ruído: nos tornamos incapazes de assimilar e tratar tanta abundância informativa. É necessário que se enfatize cada vez mais a análise da informação e que se encorajem as inovações técnicas nesse campo. Já surgem os grandes sistemas de manipulação de dados, gigantescos depósitos de dados com seus 'Data Minds', softwares usando técnicas de IA que trazem, por mecanismos de inferência, a informação desejada ou a possível informação desejada.

As possíveis reações ante esse fenômeno da 'poluição informativa' ocorreriam em três direções **[Sor92]**. Uma primeira via seria a seleção da informação, sem redundâncias nem repetições, como se mencionou algumas linhas acima. A outra seria a redução da informação, acomodando-a em função de interesses específicos e especializados do público e a terceira via é a fuga da informação. A fuga da informação seria o florescimento de ideologias simplificadoras, a semeadura do irracional, o voluntarismo irreflexivo, o empobrecimento das relações sociais e o desenvolvimento do mais passivo consumismo. É uma hipótese reducionista, somente esboçada em determinados nichos sociais. Uma futura linha de fuga seria a exploração através da venda e compra da informação, das mensagens informativas, a um determinado custo (como avaliar?), orientando-se a radiotelevisão ao volume de informação e tipo de informação que o assinante deseja.

8.2 O equilíbrio entre o toque humano e a tecnologia

As citações e considerações deste capítulo estão baseadas em uma palestra do prof. Dr. Valdemar Setzer, do Instituto de Matemática e Estatística da USP, em 03-IV-96, no Museu de Arte Contemporânea da USP.

Como citado anteriormente, o computador também exerce aquelas três ações das demais máquinas, isto é, transformar, armazenar e transportar, mas não mais matéria ou energia, e sim dados. Dados podem ser considerados abstrações do mundo real, não fáceis de se estabelecer muitas vezes. Como traduzir em símbolos uma personalidade, ou um sentimento? Para serem processados por um computador eles terão que ser tratados como símbolos formais (que já é um empobrecimento), por exemplo, introduzindo-se uma gradação numérica: tal intervalo entre números x e y representará uma gradação da intensidade de um determinado sentimento, o que é um empobrecimento ainda maior. Conforme o prof. Setzer:

"É importante fazer aqui uma distinção necessária entre o armazenamento de textos, imagens e som, que são pura e simplesmente reproduzidos, talvez com alguma edição da sua forma (por exemplo, alinhamento de parágrafos, saliência de contrastes em fotos, eliminação de ruídos), e o processamento de dados. Este último é o tratamento que se dá aos dados, transformando seu conteúdo, isto é, a semântica que se associa aos mesmos. Por exemplo, traduzir textos de uma língua para outra, extrair características de estilo de autores em pesquisas de lingüística computacional, gerar desenhos a partir de programas como no conhecido caso dos fractais, etc.".

"De onde provem o empobrecimento da informação? Do fato de que os dados não têm nada a ver com a realidade, sendo na verdade representações simbólicas de pensamentos abstratos formais, lógico-simbólicos, e como tal eles não precisam ter consistência física. Aliás, é justamente a imponderabilidade dos dados e sua alienação em relação ao físico que permitiu que os computadores fossem construídos cada vez menores, o que não pode acontecer com todas as outras máquinas. De fato, estas podem ser caracterizadas como máquinas concretas, ao passo que os computadores são máquinas matemáticas, e portanto abstratas, virtuais. Assim, todo processamento de dados deve utilizar-se exclusivamente de pensamentos formais expressos sob a forma de um programa de computador. Esse processamento lógico-simbólico, por ser extremamente restrito e unilateral, acaba por restringir enormemente o espaço de tratamento das informações, que devem ser expressas sob forma de dados; daí o empobrecimento das informações que são representadas por esses dados. Note-se que essa restrição é até de natureza matemática: não é possível colocar no computador as noções de infinito e de contínuo, apenas aproximações das mesmas".

"A caracterização do computador como máquina abstrata fica mais clara ao notar-se que todas as linguagens de programação são estritamente formais, isto é, passíveis de serem descritas matematicamente. O próprio funcionamento lógico do computador pode ser descrito por formulações lógico-matemáticas. As outras máquinas não têm essa característica, pois atuam diretamente na matéria (aí incluída a energia), e esta escapa a uma descrição matemática".

"Mas não são só as linguagens de programação que são formais. Qualquer linguagem de comandos, mesmo icônica, de um software qualquer, também é matematicamente formal. Por exemplo, qualquer comando de um editor de textos produz

uma ação do computador que é uma função matemática sobre o texto sendo trabalhado ou sobre o estado do computador. Portanto, para se programar ou usar um computador, é necessário formular os pensamentos dentro de um espaço estritamente abstrato, matemático, apesar de aparentemente não ser o tradicional, pois os símbolos e as funções são em grande parte diferentes. Programar ou usar um computador são funções estritamente matemáticas, como fazer cálculos ou provar teoremas. Assim, a programação ou uso de um computador exigem o mesmo grau de consciência e abstração que a atividade matemática. Isso não se passa com todas as outras máquinas, que exigem uma certa coordenação motora automática, semiconsciente (por exemplo, só se aprende a andar de bicicleta quando não é mais necessário pensar sobre os movimentos e o equilíbrio)".

Como uma das conseqüências dessas afirmações, pode-se propor que o uso do computador deva estar acompanhado de um novo tipo de educação, seja no âmbito da família ou das escola, das universidades ou das empresas, que aponte para uma abertura maior do entendimento humano. E esse saber vital, que faz com que um homem se sinta interiormente livre – porque tem respostas às questões da vida –, e que tenha uma visão mais ampla da realidade, é a cultura, a literatura, a filosofia, a história, etc., ou seja, as humanidades ou artes liberais, como antes eram chamadas algumas ciências humanas. Nas ciências técnicas e para os profissionais da Computação isto é mais premente. O maior problema que a especialização das ciências técnicas trouxe foi essa perda do sentido de conjunto. Continuando com as considerações da citada palestra:

"Um empobrecimento que também pode dar-se em outro sentido que é o uso da computação na arte. Há um elemento informal e intuitivo na arte que leva a dizer que na criação artística deve haver um elemento inconsciente, que nunca poderá ser conceituado totalmente. Já a criação científica deve poder ser expressa por meio de pensamentos claros, universais e não-temporais, isto é, independentes da particular interpretação do observador, talvez até certo ponto (dependendo da área) formais, matemáticos. Imagine-se uma descrição do Altar de Isenheim através dos seus pixels e seus comprimentos de onda: ele perderia totalmente o senso estético e não produziria mais a reação interior provocada no observador pelas cores, formas e motivos, isto é, não teria o efeito terapêutico para o qual foi criado por Grünewald".

"O elemento emocional foi realçado por Freud, quando afirmou em sua 'Introdução à Psicanálise,' Aula 23, e no ensaio 'Além do Princípio do Prazer,' que a arte é emoção ou expressão subconsciente e não imitação ou comunicação (dentro de seu típico raciocínio unilateral da teoria da sublimação da emoção e do desejo através da arte). Comparando-se com a arte como comunicação de uma realidade espiritual, de Kandinsky, vê-se bem o contraste entre materialismo e espiritualismo; neste pode haver algo superior a ser comunicado".

"Apesar de que a idéia expressa em um objeto de arte seja de conteúdo objetivo, a sensação e emoção que ela desperta é subjetiva. Por exemplo, ouça-se uma terça maior seguida de uma menor, ou uma sétima seguida de uma oitava. Estamos seguros que qualquer pessoa terá sensações diferentes em cada caso, que ficam claras pelo contraste entre cada intervalo e o seguinte. Mas provavelmente quase todas as pessoas dirão que a terça menor é 'mais triste' e a sétima produz uma tensão aliviada pela oitava. Cada um sente essas emoções

diferentemente, mas há claramente algo universal por detrás delas, como as sensações que temos do amarelo limão (alegre, radiante, abrindo-se) e do azul da Prússia (triste, introspectivo, fechando-se) ".

"É necessário considerar também uma distinção essencial entre obra artística e científica o fato da primeira dever sempre ter contextos temporais e espaciais ligados à sua criação. Como contraste, uma teoria científica não depende do tempo, desde que seja consistente e corresponda às observações, se for o caso. Um exemplo simples é o do conceito de uma circunferência, como por exemplo o lugar geométrico dos pontos eqüidistantes de um ponto. Essa definição formal não dependeu das condições de seu descobridor. Ela é impessoal e eterna. O fato de podermos captá-la com nosso pensamento levou Aristóteles a conjeturar, por um raciocínio puramente lógico (precursor de nossa maneira de pensar hodierna), que temos dentro de nós também algo de eterno, e que não poderia ter ocorrido em Platão, pois este tinha sido um iniciado nos Mistérios (em *A Escola de Atenas*, de Rafael há uma representação da diferença entre os dois*)".

"A dependência espaço-temporal da criação artística aliada ao elemento de expressão individual semiconsciente do artista faz com que haja sempre um elemento de imprevisibilidade na criação. O artista deve observar sua obra durante o processo de criação, para influir no mesmo e chegar a algo que não podia inicialmente prever. Isso pode ser um processo puramente interior, como no caso de um compositor que não precisa ouvir os sons de sua obra; no entanto, a sensação auditiva ao ouvi-la tocada nunca é a mesma que a que pode imaginar interiormente. Poder-se-ia argumentar que a pesquisa científica também tem elementos de imprevisibilidade. Isso pode ocorrer até na matemática: um teorema pode ser descoberto, e o seu autor ou outros ainda não saberem como se poderá prová-lo (um exemplo recente foi a prova do último teorema de Fermat, formulado no século XVII). Uma grande diferença reside no fato do resultado ser de um lado um conceito e de outro um objeto. Além disso, uma vez estabelecido um conceito científico, toda vez que se refizer a experiência ou a teoria correta o resultado será o mesmo (dentro das eventuais aproximações experimentais); no caso da criação artística, o objeto de arte deverá sempre mudar, pois a sua simples presença deve influenciar o criador, que terá outras inspirações na hora de repetir a criação (lembremos da frase de Freud de que simples imitação não é arte). Dá-se a esse fator o nome de *dinamismo da criação artística*".

"Portanto o uso do computador para fazer arte, sem considerar sua grande utilidade como banco de dados das obras artísticas, pode ser empobrecedor".

"Como instrumento passivo na criação artística, como é o caso do uso de uma ferramenta CorelDraw, existe o problema do usuário fazer uso de um raciocínio formal ao utilizar os comandos do computador submetendo a criação artística a uma consciencialização e formalização e o problema da ausência do elemento inconsciente, assim como do contato físico que desperta diferentes reações, como por exemplo, no pintor com seu pincel, no pianista ao dedilhar o piano".

---

* O autor se refere ao famoso quadro onde vários filósofos gregos aparecem, e, caminhando lado a lado, estão Platão e Aristóteles, um apontando o dedo para cima e outro para baixo, respectivamente, indicando o mundo das idéias e o mundo real.

“Uma outra forma de usar um computador em arte é fazer um programa para gerar imagens ou sons (quem sabe, no futuro, até fazer uma escultura ou construir uma casa). Um exemplo conhecido disso são os fantásticos desenhos produzidos por funções fractais; programas para produzir desenhos com essas funções provavelmente estarão logo no mercado. Nesse caso, não há apenas a substituição de um instrumento informal por outro formal; o próprio processo de criação torna-se totalmente formal. A criação deve ser expressa de maneira estritamente matemática, como é o caso de um programa. Com isso, elimina-se totalmente o elemento inconsciente. É também eliminado o elemento individual, no sentido de qualquer pessoa poder entender totalmente como a obra foi produzida - basta examinar detalhadamente o programa. É eliminado ainda o elemento temporal e espacial ligado à criação. Em outras palavras, a atividade artística tornou-se atividade científica. A propósito, é muito importante compreender-se o que significa produzir um programa para fazer uma obra de 'arte' segundo um certo estilo. Um computador pode produzir desenhos e música que se assemelham aos de Mondrian e de J.S.Bach, mas estes tiveram que desenvolver seus estilos para poder ser depois analisados e expressos em elementos puramente formais e programados em um computador, para gerar algo que aparentemente é semelhante. Sem Bach, não haveria programas que imitam sua música. Além disso, a ‘criação’ do computador não exprime nenhuma idéia além da contida no estilo, desde que este seja expresso matematicamente, o que representa um empobrecimento”.

Encerradas as considerações, o que se deseja chamar a atenção é a mentalidade do uso do computador sem o correspondente desenvolvimento de outros aspectos da inteligência do homem. Depois de tudo o que foi dito, seria redundante e supérfluo falar das vantagens desse instrumento de trabalho que potencializou e impulsionou o desenvolvimento das ciências em geral. Mas é bom lembrar que os computadores não inovam, não se relacionam, não são flexíveis e não sabem tomar iniciativas diante de situações não pré-determinadas por algoritmos internos. A imersão na informática traz o risco de se deixar de lado o cultivo ou mesmo a perda de aptidões fundamentais como a leitura, a reflexão, a criatividade, etc. Será tarefa primordial principalmente nos estabelecimentos de ensino, onde o computador se faz cada vez mais presente, preocupar-se em dar ao lado dos conhecimentos técnicos e informáticos, uma sólida formação humanística que garanta o exercício integral da inteligência humana em seus vários âmbitos.

9 Conclusão

Voltando a algumas das primeiras observações feitas no capítulo introdutório, foi dito que a exposição histórica não é apenas a narrativa de acontecimentos, cronológica e tematicamente ordenados. Esses são somente a "ponta do iceberg", pois escondem motivações, esforços, entrelaçamentos e a própria evolução anterior. A tarefa do historiador não se conclui com a obtenção de dados fidedignos, depurados e exatos, bem como o estabelecimento de séries desses fatos de maneira coerente e significativa. São somente pontos de partida para se inquirir e perguntar sobre o próprio homem, o verdadeiro protagonista da História. Esse "saber" histórico produz assim um enriquecimento da experiência humana, permite enfrentar o desafio dos novos problemas com melhores recursos. Há mais possibilidades de crescimento e criação de coisas novas quando se possui uma herança. A criatividade não se faz sobre o nada. Ao jovem que o procurou dizendo que queria fazer versos livres, Manuel Bandeira recomendou que estudasse a fundo poesia clássica, metrificada: e que só então estaria apto a fazer versos livres. A criatividade é antes extrapolar e reorganizar dados já incorporados, numa configuração nova. Mas, para extrapolar ou reorganizar dados, é preciso, antes de mais nada, tê-los.

Ao lado disso, é uma aspiração constante de qualquer cultura entender o momento presente, formar uma imagem coerente, selecionando os fatos do passado que afetaram a evolução do ser humano, que permitam construir uma explicação. Essa idéia pode ser levada também a qualquer campo do conhecimento humano e técnico. Quando se abandona o conhecimento histórico, uma ciência, uma comunidade social, o homem, ou qualquer outro âmbito, ficam privados de uma dimensão essencial na ordem do tempo: o entrelaçamento entre presente e passado em uma unidade lógica.

A Computação atravessa um tempo de expansão em várias direções, tornando-se uma tarefa necessária guardar seu patrimônio, discernindo as realidades e conceitos mais importantes. Tudo isso é importante para o ensino, pois a Computação não surgiu do nada: há uma **história** por trás de cada conceito. Cada conceito tem o seu lugar, a sua importância e a sua história que é necessário ser ensinada

Este trabalho sobre História da Computação, um entre outros que estão surgindo e alguns que já existem, faz uma retrospectiva enfatizando as idéias, os paradigmas, pretendendo apenas ser uma pontualização, visando uma futura expansão, sobre alguns aspectos abstratos do desenvolvimento dos computadores. Ele e sua futura expansão são apenas um começo, porque a área sobre a qual se falou continua em constante evolução e mudança. O campo de estudo ainda tem uma história muito recente e por demais volátil, não se podendo chegar a algo definitivo.

Apesar dessas dificuldades, deve-se continuar a chamar a atenção sobre a importância de se registrar e estudar o desenvolvimento dos computadores eletrônicos e a conseqüente evolução dos temas anexos: Linguagens de Programação, a Teoria da Computação, estudo da Complexidade dos Algoritmos, etc., assim como a importância decisiva do fator humano. Quando tantos se maravilharam com a derrota do campeão mundial de xadrez Kasparov para o computador da IBM, o *Deep Blue,* (abril/maio de 1997), surpreende a pouca atenção dada à equipe de técnicos que construiu e programou a máquina,

às heurísticas utilizadas e aos objetivos que estão por detrás desse novo engenho, como se alguém ficasse maravilhado com o quadro e os pincéis de uma obra de arte e se esquecesse do artista. A história tem o dom de focalizar com especial nitidez aquele que é o seu personagem principal: o homem.

10 Referências bibliográficas

[AG81] Anawati, M-M & Gardet, Louis. *Introduction a la Théologie Musumane*. Paris: Vrin, 1981, apud **[Lau97]**.

[Aga86] Agazzi, Evandro. *La logica simbolica.* Barcelona: Herder, 1986.

[Ara78] Aracil, J. *Introducción a la dinámica de sistemas*. Madrid: Alianza, 1978, **apud [Tir2002]**.

[Arb87] Arbib, Michael A. *Brain, Machines and Mathematics.* New York: Spring-Verlag, 1987.

[Art94] Artigas, Mariano. *El desafio de la racionalidad.* Barcelona: EUNSA, 1994.

[Asp90] Aspray, Willian. *John von Neumann and the origins of modern computing*. New Baskerville: Massachusetts Institute of Technology, 1990.

[Bar67] Barker, Stephen F. *Filosofia da matemática*. Rio de Janeiro: Zahar, 1969.

[Bar84] Barendregt, H.P. *The lambda calculus, its syntax and semantics.* Amsterdan: Elsevier Science Publishers B.V., 1984.

[BBF82] Berg, H.K. & Boebert, W.E & Franta, W.R. & Moher, T.G. *Formal methods of program verification and specification.* New Jersey: Prentice-Hall, Inc., 1982.

[Bel37] Bell, E. T. *Men of Mathematics.* New York: Simon & Schuster, Inc, 1937.

[BL74] Brained, W. & Landweber. *Theory of computation.* New York: Wiley-Interscience Publication, 1974.

[Boc66] Bochenski, J. M. *Historia de la logica formal.* Madrid: Biblioteca Hispânica de Filosofia, 1966.

[Bol84] Bolter, J. David. *Turing's man, western culture in the computer age.* Carolina do Norte: Universidade da Carolina do Norte, 1984.

[Boo84] Boole, George. *El analises matematico de la lógica.* Madrid: Catedra, 1984.

[Bow94] Bowen, Jonathan P. *A brief history of algebra and computing: an ecletic oxonian view.* Oxford: Librarian, Oxford University Computing Laboratory, 1994.

[Boy74] Boyer, Carl B. *História da Matemática.* São Paulo: Edgard Blucher, 1974.

[Bri79a] Encyclopedia Britannica *The New, Macropaedia*. USA: Helen Hemingway Benton, 1979.

[Bri79b] Encyclopedia Britannica *The New, Micropaedia*. USA: Helen Hemingway Benton, 1979.

[Bur51a] Burks, Arthur W. [1951]. *An intermediate program language as an aid in program synthesis*. Engineering Research Institute, Report for Burroughs Adding Machine Company, 1951.

[Bur51b] Burks, Arthur W. [1980]. *From de ENIAC to the stored-program computer,* in *A history of computing in the twentieth century (a collection of essays)*. London: Academic Press, 1980.

[Bur92a] Burke, Peter. *A escola dos annales, 1929-1989-A revolução francesa da historiografia*. São Paulo: Editora UNESP, 1992.

[Bur92b] Burke, Peter. *A escrita da história-novas perspectivas*. São Paulo: Editora UNESP, 1992.

[Cas97] Casti, John L. *Five golden rules-great theories of 20th-century Mathematics and why they matter*. Canadá: John Wiley Sons, Inc., 1997.

[CO98] Carvalho, Roberto Lins de & Oliveira, Cláudia Maria G.M. de. *Modelos de computação e sistemas formais*. Rio de Janeiro: DCC/IM, COPPE/Sistemas, NCE/UFRJ, 11ª Escola de Computação, 1998.

[Coe95] Coelho, Clarimar J. *Monografia: apresentação das bases teóricas para o desenvolvimento de um sistema de apoio a decisão utilizando resolução de equações booleanas*. Goiás: Depto. Matemática e Física da Universidade de Goiás, 1995.

[Cof91] Coffa, J. Alberto Wessels. *The semantic tradition from Kant to Carnap*. Cambridge: Cambridge University Press, 1991.

[Coh87] Cohen, Daniel E. *Computabilty and logic*. England: Ellis Horwood Series in Mathematics and Its Aplications, 1987.

[Cos77] Costa, Newton C. A. *Introdução aos fundamentos da Matemática*. São Paulo: Editora Hucitec, 1977.

[Dan54] Dantzig, T. *Number, the language of science*. New York: Doubleday Anchor Books, 1954.

[Die81] Dieudonné, J. *Von Neumann, Johann (or John)*, in Gillespie, Charles C., *Dictionary of Scientific Biography*. New York: Charles Scribner's Sons, 1981.

[Fer85] Fernandez, Luiz Soares. *Historia Universal*, Volume 1. Pamplona: Edições Universidade de Navarra, 1985.

[Gat95] Gates, Bill. *A estrada do futuro.* São Paulo: Companhia das Letras, 1995.

[GM95] Gosling, James & McGilton, Henry [1995]. *The Java language environment- a white paper.* California: Sun Microsystems, Inc, 1995.

[GN88] Le Goff, Jacques & Nora, Pierre. *História: novos problemas.* Rio de Janeiro: Livraria Francisco Alves Editora, 1988.

[Gof94] Le Goff, Jacques. *História e memória.* Campinas: Editora da UNICAMP, 1994.

[Gol72] Goldstine, Herman H. *The computer from Pascal to von Neumann.* New Jersey: Princeton University Press, 1972.

[Gom97] Gomes, Nelson Gonçalves. *Lógica Matemática*, Curso de Lógica da Faculdade de Filosofia da UnB, apostila para apontamentos de aula, 1997.

[H30] H.W.B., Joseph [1916|1930]. *Introduction to Logic.* Oxford: Clarendon Press, 1930.

[Har96] Harrison, John, *The history of formal logic.*. Revista Mathesis Universalis, 1996, nº 2, Formalized Mathematics. Disponível na internet http://saxon.pip.com.pl/MatUniversalis/2/harrison/jrh03.html.

[Her69] Hermes, Hans. *Enumerability, decidability, computability-an introduction to the theory of recursive functions.* New York: Springer-Verlag, 1969.

[Hod83] Hodges, Alan. *Alan Turing: the Enigma.* New York: Simon & Schuster, 1983.

[Hur80] Hurd, Cuthbert. *Computer Development at IBM,* in *A history of computing in the twentieth century (a collection of essays).* London: Academic Press, 1980.

[IEEE95] IEEE Computer Society, *Edição comemorativa dos 50 anos da revista*. Disponível na Internet, http://www.computer.org/50.

[Ifr89] Ifrah, Georges. *Os números, a história de uma grande invenção.* Rio de Janeiro: Globo, 1989.

[Kah67] Kahn, D. *The Code Breakers.* New York: Macmillan, 1967.

[Kar61] Karlson, Paul. *A magia dos números.* Rio de Janeiro: Editora Globo, 1961.

[Kne68] Kneale, Willian & Kneale, Martha *O desenvolvimento da lógica.* Lisboa: Fundação Calouste Gulbenkian, 1968.

[Knu69] Knuth, D. E. *The Art of Computer Programming, volume 1.* Massachusetts: Addison-Wesley, 1969.

[Knu76] Knuth, D. E. *Ancient babylonian algorithms.* In Comm. ACM 15, 671-677(1972); Errata, Comm. ACM 19, 108, 1976.

[Kow96] Kowaltowski, Tomasz. *Von Neumann: suas contribuições à Computação,* in Estudos Avançados 10(26). Campinas: UNICAMP, 1996.

[KP80] Knuth, D. E. & Pardo, L. T. *The early development of programming languages*, in *A history of computing in the twentieth century (a collection of essays).* London: Academic Press, 1980.

[Lau97] Lauand. Luis Jean. *Ética e antropologia: estudos e traduções*. São Paulo: Mandruvá, 1997.

[Lee95] Lee, J. A. N., *Relatório preparatório sobre o estudo da História da Computação, e propostas preliminares para inclusão de aspectos da História da Computação curriculum das universidades e colégios, apresentado no Grupo de Trabalho 9.7 (História da Computação), do IFIP* (International Federation for Information Processing), 24 a 29 de julho de 1995.

[Lee96] Lee, John A. N. *Annals of the History of Computing,* Volume 18, nº 2. EUA: Institute of Electrical and Electronics Engineers, Inc., 1996.

[Lil45] Lilley, S. *Machinery in Mathematics, a historical survey of calculating machines*. London: Discovery, 1945.

[Lin96] Linden, Peter van der. *Just Java.* California: The Sunsoft Press, 1996.

[LSSK79] Lucchesi, C. L. & Simon, I. S. I. & Simon, J. & Kowaltowski, T. *Aspectos Teóricos da Computação.* Rio de Janeiro: Livros Técnicos e Científicos S.A., 1979.

[Luc82] Lucena, Carlos J.P. *Análise e síntese de programas de computador*. Brasília: Editora Universidade de Brasília, 1982.

[May80] May, Kenneth O. *Historiography: a perspective for computer scientists,* in *A history of computing in the twentieth century (a collection of essays).* London: Academic Press, 1980.

[Moo77] Moore, Doris Langley. *Ada, countess of Lovelace: Byron's legitimate daughter.* New York: Harper and Row, 1977.

[Mor61] Morrison, Philip & Morrison, Emily. *Charles Babbage and his calculatings engines.* New York: Dover, 1961.

[Mos92] Mosses, Peter D. *Action Semantics.* Cambridge: Cambridge University Press, 1992.

[Mot96] Motoyama, Shozo. In *História da técnica e da tecnologia no Brasil.* São Paulo: UNESP, 1996.

[MP43] McCulloch , Warren & Pitts, Walter. *A logical calculus of the ideas immanent in nervous activity,* Bulletin of Mathematical Biophysics, volume 5, 1943.

[MS98] Michelson, A..A.. & Stratton, S.W. *A New Harmonic Analyzer.* American Journal of Science, 4ª série, vol V, 1898.

[Nau69] Naur, P. Randell B. *Software Engineering.* Rept. NATO Sci. Comm., 1969.

[Nee59] Needham, Joseph. *Science and civilization in China*, volume 3. Cambridge: Cambridge University Press, 1959.

[New56] Newman, J. R. *The world of mathematics.* New York: Simon & Schuster, 1956.

[NN56] Nagel, Ernest & Newman, J. R. "*Gödels Proof", the world of mathematics.* New York: Simon e Schuster, 1956.

[Per88] Pérez, Rafael Gomes. *História básica da filosofia.* São Paulo: Nerman, 1988.

[RA91] Realle, G. & Antisseri, Dario. *História da Filosofia.* São Paulo: Edições Paulinas, 1991.

[Rhe85] Rheingold, Howard. *Tools for thought: the people and Ideas of the next computer revolution.* New York: Simon & Schuster, 1985.

[RN95] Russell, Stuart & Norvig, Peter. *Artificial Intelligence, a modern approach.* New Jersey: Prentice Hall Series in Artificial Intelligence, 1995.

[Roc81] Roces, Wenceslao. *Lecciones sobre la historia de la filosofia, III.* (trad. *Vorlesungen über die Geschichte der Philosophie.* Karl Ludwig Michelet, 1842). México: Fondo de Cultura Económica, 1981.

[Ryd67] Ryder, John D. *Engineering Electronics with Industrial Applications and Control.* Toquio: McGraw-Hill Kogakuska, Ltd., 1967.

[San82] Sanguineti, Juan Jose. *Lógica.* Pamplona: EUNSA, 1982.

[Sch31] Scholz, H.. *Abriss der Geschichte*, apud **[Aga86]**, pag. 61

[Sho67] Shoenfield, Joseph R. *Mathematical logic.* Massachusetts: Addison-Wesley Publishing Company, 1967.

[Sin99] Singh, Simon. *O último teorema de Fermat.* Rio de Janeiro: Record, 1999.

[Sho71] Shoenfield, Joseph R. *Degrees of Unsolvability.* Amsterdam: North-Holland Publishing Company, 1971.

[Ste91] Steiner, George. *Presencias reales.* Barcelona: Destino, 1991.

[Sor92] Soria, Carlos. *De la metáfora del cuarto poder a la sociedad de la información*, in Veinte claves para la nueva era. Madrid: Ediciones Rialp S/A, 1992.

[Sti80] Stibitz, George R. *Early computers,* in *A history of computing in the twentieth century (a collection of essays).* London: Academic Press, 1980.

[Tan92] Tanenbaum, A. S. *Organização Estruturada de Computadores.* Rio de Janeiro: Prentice-Hall do Brasil, 1992.

[Tei97] Teixeira, João de Fernandes Teixeira *Mentes e Máquinas.* Porto Alegre: Artes Médicas, 1997.

[Tir2002] Andrés, Tirso de. *Homo cybersapiens, la inteligência artificial y la humana.* Pamplona: EUNSA, 2002.

[Toy87] Toynbee, Arnold. *Um estudo da história.* Brasília: Martins Fontes & Editora Universidade de Brasília, 1987.

[Tur36] Turing, A. M., *On computable numbers, with an aplication to the Entscheidungsproblem.* Proceedings of the London Mathematical Siciety, volume 42, Londres, 1936.

[Tur50] Turing, A. M. *Computing machinery and intelligence.* England: Mind, volume 59, nº 236, 1950.

[Ula80] Ulam, S. M. *Von Neumann: The Interaction of Mathematics and Computing,* in *A history of computing in the twentieth century (a collection of essays).* London: Academic Press, 1980.

[Vey82] Veyne, Paul M. [1982]. *Como se escreve a história.* Brasília: Editora Universidade de Brasília, 1982.

[Wat90] Watt, David A.. *Programming language concepts and paradigms.* Cambridge: Prentice Hall, 1990.

[Wex80] Wexelblat, R. L. [1980]. *ACM History of programming language conference, Los Angeles.* New York: ACM Monograph, Academic Press, 1980.

[Wie70] Wiener, Norbert. *Cibernética.* São Paulo: Editora da Universidade de São Paulo e Editora Polígono, 1970.

[Wil65] Wilder, Raymond L. *Introduction to the foundations of Mathematics.* Tóquio: John Wiley & Sons, Inc., 1965.

[Wil97] Willians, Michael R. *History of computing technology.* California: IEEE Computer Societ Press, 1997.

[Zus80] Zuse, Konrad [1980]. Some remarks of the history of computing in Germany, in *A history of computing in the twentieth century (a collection of essays).* London: Academic Press, 1980.

Anexo – Cronologia (até o ano 2007)*

DATA	EVOLUÇÃO CONCEITUAL	EVOLUÇÃO TECNOLÓGICA
4000 aC		Registros de transações comerciais em pequenas tábuas
3000 aC		Provável aparecimento do ÁBACO
1800 aC	Babilônia, métodos para resolver problemas numéricos	
1650 aC	Rhind Papyrus: palavra álgebra (al-jabr, reunião de partes separadas)	
500 aC		Egito, ábaco com fios
384 aC	Aristóteles: início da ciência lógica (Organon, cj. de obras editadas posteriormente: lógica formal e filosófica)	
330 aC	Euclides, em sua obra Elementos, método axiomático: postulados, proposições, teoremas	
250 aC	Crivo de Eratóstenes para números primos	
79 aC		Dispositivo "Antikythera", para cálculo de calendário lunar
830	Abu Ja'far Muhammad ibn Musa al'Khwarizmi, álgebra	
1000		Gerbert de Aurillac ou Papa Silvestre II, ábaco mais eficiente
1300	Raymundus Lullus, Ars magna: o 1º dispositivo de texto ocidental para produção de sentenças logicamente corretas	
1445		al-Kashi: Tabac al-Manatec - dispositivo para simplificar cálculos de importantes tempos associados aos eclipses lunares
1614	John Napier, ponto decimal, tabela de logaritmos dispositivo que usa ossos para demonstrar a divisão através de subtrações e	John Napier, dispositivo que usa ossos para demonstrar a divisão através de subtrações e multiplicação por adições

* Esta cronologia vai até o ano 2007 somente, e não sei se serão feitas novas atualizações, ao menos por esse autor. A história segue, a tecnologia avança, talvez novos paradigmas alterem rotas, enquanto nós, pobres homens, ficamos... Espero que os interessados no assunto prossigam com a divulgação de novos aspectos ou aprofundamentos da História da Computação, proporcionando também melhores tabelas (e mais atualizadas!).

	multiplicação por adições	
1622		William Oughtred, régua de cálculo circular baseada nos logaritmos de Napier
1624		Wilhelm Schickard, "relógio de calcular" para multiplicação de grandes números
1642		Blaise Pascal, 1ª máquina numérica de calcular
1666	Leibniz: De Arte Combinatoria; 'characteristica universalis	
1673		Leibniz, dispositivo mecânico de calcular que multiplica, divide, soma e subtrai
1750		Artesão suiço cria autômatos com mecanismos de trabalho temporais para bater teclas e escrever letras
1780		Benjamin Franklin, eletricidade.
1801		Joseph-Marie Jacquard, cartões perfurados para automatizar seus teares
1814		J.H.Herman, primeiro planímetro, dispositivo analógico para medir área coberta por uma curva em um gráfico
1822	Charles Babbage, projeto Engenho Diferencial para calcular logaritmos	
1829		Willian Austin Burt, EUA, 1ª máquina de escrever
1833	Charles Babbage, projeto Máquina Analítica, cartões perfurados: 1º modelo teórico de um computador.	
1838	Samuel Morse e Alfrdd Vail demonstram os elementos do sistema de telégrafo	
1842	Lady Ada Byron, 1º programa para a máquina de Babbage	
1847	George Boole, The Mathematical Analysis of Logic - nasce a Lógica Simbólica; sistema binário	
1854	George Boole, An Investigation of the Laws of Thougt	
1855		James Clerk Maxwell: planímetro rotacional
1855		George e Edvard Scheutz de Estocolmo, 1º computador mecânico, baseados no trabalho de Babbage.
1858		Jako Amsler: planímetro polar (pré-computador

		analógico)
1876		O telefone é inventado por Alexander Graham Bell
1878		Sir Willian Thomson (Lord Kelvin): analisador harmônico (pré-computador analógico)
1879	Gottlob Frege: Begriffsschrift (Ideografia ou Conceitografia) - rigor formal	
1874- 97	Cantor - desenvolvimento da Teoria dos Conjuntos	
1886		William Burroughs, 1ª máquina mecânica de calcular
1889		Herman Hollerith, patente de máquina tabuladora
1890		Herman Hollerith, máquina eletromecânica, cartões perfurados, censo EUA
1893	Gottlob Frege: Grundgesitze (Leis Fundamentais da Aritmética, Ideograficamente deduzidas) - redução da aritmética à logica, teoria da linguagem	
1895		Guglielmo Marconi transmite um sinal de rádio
1900	Hilbert, 10º problema: existe procedimento de decisão para solucão das equações diofantinas?	Hollerith funda a Tabulating Machine Co. e constrói um dispositivo classificador
1903		Nikola Tesla, patenteia um circuito lógico eletrico chamado porta ou chave
1904		John A. Fleming patenteia um tubo de diodo a vácuo (válvula)
1908	G. Peano: Formulário Matemático, simbolismo conectado com estrutura das linguagens naturais	
1913	Bertrand Russell e Whitehead: Principia Mathematica, deduções matemáticas a partir do cálculo lógico	
1911	Kamerlingh Onnes, físico, na Universidade de Leiden descobre a supercondutividade	
1911		Computer-Tabulating-Recording Company é formada da união de Tabulating Company, Computing Scale Company, e International Time Recording Company

1915		O uso de microchips é prefigurado pelo físico Manson Benedicks que descobre que o cristal de germânio pode ser usado para converter corrente alternada para corrente direta
1919		Eccles e Jordan, físicos americanos, inventam o chaveamento eletrônico flip-flop, crítico para altas velocidades
1921	A palavra “robot” é usada pela 1ª vez por Karel Câpek, em seu trabalho Rossum’s Universal Robots	
1924		Computing-Tabulating-Recording Company muda seu nome para International Business Machines
1925		Vannevar Bush, inicia a construção de dispositivo para resolver equações diferenciais, em MIT
1927		Radio-telefonia torna-se operacional entre Londres e Nova Iorque
1927		Powers Accounting Machine Company torna-se Tabulating Machines Division da Remington-Rand Corp.
1928		Aparece o relógio de cristal de quartzo
1928		Vladimir Zworykin, inventa o tubo de raios catódicos
1930		O Analisador Diferencial, inventado por Vannevar Bussh e colegas, no MIT, resolve inúmeras equações diferenciais
1930	Gödel: Teorema da Incompletude	
1931		Reynold B. Johnson, professor em Michigan, inventa um processo de marcar em uma folha de respostas através de uma caneta sensível à condutividade. A IBM comprou mais tarde esta tecnologia
1931		1º computador mecânico é construido na Alemanha, por Konrad Zuse
1933		1ª máquina eletrônica que se comunica, Voder, é construída por Dudley, que seguiu em 1939 com o Vocoder (codificador de voz)
1936	Konrad Zuse compreende que programas compostos de combinações de bits podem	

	ser armazenados	
1936	Alonso Church: funcões computáveis, indecidibilidade da lógica de 1ª ordem	
1936	Alan M. Turing, Universidade de Princeton: computabilidade e Máquina de Turing	
1937		George Stibitz, 1º circuito binário baseado na álgebra booleana, Bell Telephone Laboratories
1937	Claude Shannon: princípios para um somador eletrônico de base 2	
1937	Howard Aiken submete à IBM proposta de máquina calculadora digital, capaz de fazer as 4 operações fundamentais e operar mediante instruções sequenciais	
1937	John Vincent Atanasoff elabora os princípios para um computador eletrônico digital	
1938		Zuse completa o Z1, computador eletromecânico binário e o refina desenhando o Z2
1938		Hewlett-Packard Co. fundada para fazer equipamentos eletrônicos
1938	Isaac Asimov: termo "robot"	
1939		Trabalhando de outubro a novembro, John Vincent Atanasoff com Clifford E. Berry, controem um protótipo de computador eletrônico digital que usa aritmética binária
1940		Bell Labs, George Stibitz, Calculador de Números Complexos, computador digital
1940		TV a cores
1940		Bell Laboratories, 1º terminal.
1940		Konrad Zuze completa o Z2
1941	Colossus é projetado por Alan M. Turing e iniciada a sua construção por M.H.A. Neuman e Tommy Flowers, Universidade de Manchester, 1º dispositivo de calcular eletrônico(participação de Alan Turing)	
1941		Konrad Zuse constrói o computador Z3 a 1ª máquina de calcular com controle automatico de suas operações
1943		31-V, começa a construção do ENIAC, na Moore School of Electrical Engineering, Filadelfia

1943		Dezembro, Colossus, torna-se operacional em Bletchley Park
1943	Post: 1º sistema gerativo para computação simbólica	
1944		Mark I (IBM Automatic Sequence Controlled Calculator) é terminado pelo prof. Howard H. Aiken em Harvard junto à IBM: baseado em relês
1944	Grace Murray 1º programador do Mark I	
1945		Z4 de Zuze sobrevive à II Guerra
1945	J. Presper Eckert e John Mauchly assinam contrato para construir o EDVAC (Electronic Discrete Variable Automatic Computer). John von Neumann introduz o conceito de programa armazenado, no rascunho do projeto do EDVAC	
1945		Na primavera deste ano ENIAC esá pronto e executando
1945		Trabalhando em um protótipo do Mark II, Grace Murray Hopper encontra o 1º "bug," uma mariposa que causou uma falha em um dos relês
1946		Binac (Binary Automatic Computer), computador para operar em tempo real, iniciado por Eckert and Mauchly; completado em 1949
1946	Wiener: cibernética	
1946		ENIAC (Electronic Numerical Integrator and Computer), J. Presper Eckert e John Mauchly, 18.000 válvulas, Universidade de Pensilvânia. 5.000 adições e 360 multiplicações por segundo
1946	Arthur Burks, Herman Goldstine, e John von Neumann escrevem "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument"	
1946		Eckert-Mauchly Computer Corporation é constituida como Electronic Control Co., Universal Automatic Computer (Univac).
1946	John Tukey, conceito de bit	
1947	Alan M. Turing, artigo sobre Máquinas Inteligentes, início da IA	
1947	Association for Computing Machinery	

	(ACM) é constituida	
1947		Howard Ayken e equipe completam o Harvard Mark II
1948		EDSAC (Electronic Delay Storage Automatic Calculator), na Universidade Cambridge, por Maurice V. Wilkes
1948		IBM introduz o computador eletrônico 604
1948		IBM constrói o Selective Sequence Electronic Calculator (SSEC), computador com 12,000 válvulas
1948		Invenção do Transistor, William Bradford Shockley e John Bardeen e Walter H. Brattain
1948		Manchester Mark I, ou "baby machine", computador digital operacional, com programa armazenado
1948	Claude E. Shannon publica "A Mathematical Theory of Communication", formulando as bases para uma moderna compreensão dos processos de transmissao de informação	
1948	Richard Hamming encontra e corrige erros em blocos de dados. O código Hamming é usado posteriormente em computadores e chaveamentos telefônicos	
1949		EDVAC (Electronic Discrete Variable Automatic Computer) é testado com os 1ºs discos magnéticos
1949		Computador Whirlwind, MIT: 1º computador de tempo-real, Jay Forrester e Ken Olsen
1949		EDSAC (Electronic Delayed Storage Automatic Computer), computador com programa armazenado, Maurice Wilkes, Universidade de Cambridge, faz seu 1º calculo dia 6 de maio
1949	Short Order Code, desenvolvido por John Mauchly, a 1ª linguagem de programação de alto nível	
1949	Claude Shannon inventa a 1ª máquina de jogar xadrez	Claude Shannon inventa a 1ª máquina de jogar xadrez
1949		Jay Forrester inventa a memória de núcleos magnéticos
1950	Maurice V. Wilkes, universidade de	

	Cambridge, usa uma linguagem simbólica de montagem(assembler) no EDSAC	
1950		SEAC (Standards Eastern Automatic Computer) é desenvolvido para o National Bureau of Standards
1951		William Shockley inventa o transistor de junção
1951	1ª conferência internacional sobre computadores	
1951	Maurice V. Wilkes, conceito de microprogramação	
1951	IEEE Computer Society é constituida	
1951		UNIVAC I é instalado no Bureau of Census americano, usando fita magnética como um buffer de memória
1952	1º manual de computador, Fred Gruenberger.	
1952	Kleene: teorema da forma normal, funções recursivas	
1952		Computador IAS (von Neumann): a maioria das máquina atuais utiliza este projeto
1952		IBM 701
1952		Nixdorf Computer ié fundada na Alemanha
1952		RCA desenvolve o Bizmac, com memória de núcleos magnéticos e um tambor magnético para o 1º banco de dados
1953		1º computador da IBM para grandes massas de dados: utilizando tambores magnéticos
1953	Burroughs Corp. instala o Universal Digital Electronic Computer (UDEC) na universidade de Wayne State	
1953		Remington-Rand, para uso no Univac, 1ª impressora de alta velocidade
1953		1º dispositivo de fita magnética, o IBM 726, 100 caracteres-por-polegada de densidade e 75 polegadas por segundo de velocidade
1954		Earl Masterson's Uniprinter, ou impressora de linha, desenvolvida para computadores, 600 lpm
1954	FORTRAN é desenvolvido por John Backus, IBM. Harlan Herrick executa com sucesso o 1º programa em FORTRAN	
1954		Gene Amdahl, 1º sistema operacional, IBM 704

1954		Univac 1103ª: a 1ª máquina comercial com memória de núcleos de ferrite
1956	APT (Automatic Programmed Tool) é desenvolvido por D.T. Ross	
1956	Edsger Dijkstra: algoritmo eficiente para caminhos curtos em grafos e minimizar expansão de árvores	
1956	A. Newell, D. Shaw e F. Simon inventam o IPL (Information Processing Language.)	
1956	"Logic Theorist": dispositivo baseado em IA capaz de provar prpoposições lógicas (Dartmouth College)	
1956	John McCarthy e Marvin Minsky reúnem-se em Dartmouth College onde o conceito de inteligência artificial é desenvolvido	
1956	O termo Inteligência Artificial é usado por John McCarthy.	
1957		Control Data Corporation é formada por William C. Norris e grupo de engenheiros da Sperry-Rand
1957	Ford e Fulkerson: avanços na combinatória e algoritmos eficientes para cáculo de fluxo máximo em redes	
1957		Digital Equipment Corporation é fundada por Ken Olsen
1957	John Backus e colegas da IBM desenvolvem o 1º compilador FORTRAN para a Westinghouse	
1958	ALGOL, primeiramente chamado IAL (International Algebraic Language), é apresentado em Zurich	
1958		A 1ª máquina de memória virtual, Atlas, é instalada na Inglaterra por Ferranti. Desenvolvida ne Universidade de Manchester por R.M. Kilburn
1958		1ºs computadores eletrônicos no Japão: NEC-1101 e -1102
1958		Frank Rosenblatt contrói o Perceptron Mark I usando uma CRT (character caracter recognation) como dispositivo de saída

1958	LISP, IBM 704, MIT, John McCarthy: processamento de listas, recursividade, lambda calculus, aceleração pesquisas IA	
1958		Seymour Cray constrói o 1º supercomputador totalmente transistorizado para a Control Data Corp., o CDC 1604
1958-1959	N. Chomski, M. Rabin: especificações formais	
1958		"Ultimate", 1º robot, F.Engleberger e George C. Devol, na Gen.Motors, controlar máq. térmicas
1958		Jack Kilby da Texas Instruments elabora o 1º circuito integrado
1959	COBOL é definido pela Conference on Data System Languages (Codasyl)	
1959		IBM monta seus 1ºs computadores transistorizados: 1620 e 1790
1959	Jack S. Kilby na Texas Instruments arquiva uma patente para o 1º circuito integrado	
1959	Robert Noyce da Fairchild Semiconductor desenvolve a idéia de um bloco de circuitos integrados	
1960		Benjamin Curley, 1º minicomputador, o PDP-1, na Digital Equipment Corporation
1960		Control Data Corporation, CDC 1604, 1º computador científico
1960	Métodos heurísticos para problemas intratáveis: Karp e Kernigan	
1960	Hoare, Dijkstra, Bobo Floyd: métodos formais para corretude de programas	
1960	Inst.Pesq.Stanford & Univ. Edimburg: equipe de IA para projetar robot com visão	
1960	Algol 60 é desenvolvido por cientistas da computação americanos e europeus	
1961	O conceito de multiprogramação é implementado no IBM 7030. Time-sharing é implementado no MIT no IBM 709 and 7090 por F. Corbato.	
1961		IBM desenvolve o IBM 7030 para Los Alamos: transistorizado, 64-bit data paths, 1º de byte de 8 bits; em uso até l971
1962	APL (A Programming Language), Ken	

	Iverson, Universidade de Harvard University e IBM	
1962	Linguagens de simulação de uso geral: (1) SIMSCRIPT, por Rand Corporation; (2) GPSS por IBM	
1962		IBM 1311 discos removíveis
1962		H. Ross Perot funda EDS (Electronic Data Systems), Dallas, TX.
1963	Código ASCII padrão para troca de informações entre computadores	
1963		B5000 (Burroughs): primeira máquina projetada para uma linguagem de alto nível
1963		Consoles gráficas, General Motors (DAC-1) e MIT Lincoln Laboratories (Sketchpad), resultando em computer-aided design (CAD). 1ª light-pen, desenvolvida por Ivan Sutherland
1963		DEC, PDP-5 minicomputador
1964		IBM System 360, 1ª família de computadore compatíveis (circuíto integrado)
1964		Control Data Corporation, CDC 6000, palavra de 60 bits, processamento paralelo. CDC 6600, o mais poderoso computador por longos anos. Projetado por Seymour Cray
1964	BASIC (Beginners All-purpose Symbolic Instruction Language) é criado por Tom Kurtz e John Kemeny of Dartmouth	
1965	Universidade de Belgrado, Rajko Tomovic faz uma das 1ª tentativas de desenvolver um mecanismo artificial sensível ao toque	
1965	Robin M., Yamada, Edmond, Hartman & Stearns: Teoria da Complexidade Computacional	
1966		Texas Instruments oferece a 1ª calculadora de mão baseada em estado sólido
1967		A.H. Bobeck, na Bell Laboratories desenvolve a 1ª memória de bolha magnética

1967	Ole-Johan Dahl e Kristen Nygaard, no Norwegian Computing Centre terminam uma versão da linguagem de uso geral Simula, a 1ª orientada a objeto	
1968-1973	Donald Knuth: The Art of Computer Programming, algoritmos e estruturas de dados como entidades separadas dos programas	
1968	Edsger Dijkstra escreve sobre programação estruturada	
1968	Dendral, 1º programa de diagnóstico médico, Joshua Lederberg, Universidade de Stanford	
1969		Edson de Castro deixa a DEC to começar a Data General Corp. e introduz o Nova, 1º minicomputador de 16 bits
1969	1ª Confer. Internacional sobre Inteligência Artificial	
1969	IBM "abre" hardware e software; introduz linha de minicomputador, Sistema /3	IBM "abre" hardware e software; introduz linha de minicomputador, Sistema /3
1969	Complilador PASCAL, by Nicklaus Wirth, instalado no CDC 6400	
1970		Shakey, desenvolvido na SRI International, é o 1º robot que usa inteligência artificial para se deslocar
1970		DEC, PDP-11/20, 1º mini de 16 bits
1970	1º torneio de xadrez entre máquinas da ACM	
1970		IBM monta o 1º Sistema 370, computador de 4ª geração
1971		Floppy disks são introduzidos para carregar o microcódigo do IBM 370
1971	Cook, Levin: problemas NP-completos	
1971		Intel Corporation anuncia o 1º microprocessador, o Intel 4004, equipe liderada por Marcian E. Hoff.
1971		John Blankenbaker, 1º computador pessoal, Kenbak I
1972		1ª calculadora eletrônica, por Jack Kilby, Jerry Merryman, e Jim VanTassel, da Texas

1972	Gary Kildall, Naval Postgraduate School escreve o PL/1, a 1ª linguagem de programação para o microprocessador Intel 4004	
1972	SIRCH, dispositivo capaz de reconhecimento e orientação, Universidade de Nottingham	
1972		Intel 8008, microprocessador de 8 bits
1972	Dennis Ritchie desenvolve a linguagem C nos Bell Labs	
1972	Teorias de análise de complexidade desenvolvem a idéia da incompletude dos problemas NP, mostrando a intratabilidade de determinados problemas computacionais como o do caixeiro viajante	
1973	Linguagem PROLOG, Alain Comerauer, Universidade de Marseilles-Luminy, França	
1973		R2E comercializa o MICRAL, 1º microcomputador da França
1973		Winchester disk drives, pela IBM, nome usado para seu dispositivo de acesso direto à memória, Modelo 3340
1974		Intel 8080, microprocessador de 8 bits, usado em muitos microcomputadores
1974		Zilog é formada
1975		Cray-1, supercomputador
1975		Impressora laser (IBM)
1975		MITS, computador pessoal Altair, Ed Roberts and Bill Yates.
1975		Cincinnati Milacron T3, 1º robot da indústria aeroespacial
1975		Microsoft é formada após Bill Gates e Paul Allen adaptarem e venderem um BASIC para o Altair PC da MITS
1976		Impressoras jato de tinta (IBM)
1976	MYCIN, expert system, Stanford, por E. Shortliffe	
1976		NEC System 800 e 900, mainframes
1976		Seymour Cray desenvolve o Cray 1 com 100 milhões de operações de ponto flutuante por segundo (MFLOP)

1976	Kernigan: algoritmos aproximativos; Solovai e Strassen: algoritmos randomizados	
1976		Superminicomputadores são apresentados por Perkin-Elmer e Gould SEL.
1976		Zilog Z-80 chip
1977		Apple Computer é formada e apresenta o computador pessoal Apple II
1977		DEC, supermini VAX-11/780, 32 bits
1977		Datapoint , sistema ARC system, a 1ª área de rede local
1978	Popularização da análise estruturada (Tom de Marco)	
1978	Texas Instruments, brinquedo educacional Speak-and-Spell, síntese de voz	
1978	1ª COMDEX	
1979	Linguagem Ada, CII-Honeywell Bull (França), Jean Ichbiah	
1979	Benoit Mandelbrot continua sua pesquisa sobre fractais, gerando o conjunto Mandelbrot, derivado de $z(n + 1) = z(n) * z(n) - (0)$	
1979		VisiCalc, software de planilha eletrônica
1979		Wordstar, software de processamneto de texto, Micropro (agora Wordstar International).
1980		Control Data Corporation, Cyber 205 supercomputador
1980	Análise probabilística de algoritmos	
1980	Microsoft licencia o sistema operacional UNIX, da Bell Laboratories e aprsenta sua adaptação, o XENIX	
1980-90	Criptografia: protocolos de chave pública, baseados na geração de números primos	
1981		Commodore, VIC-20 computador para uso doméstico
1981		IBM entra no mercado de computadores pessoais
1981		Osborne Computer, Osborne 1, o 1º laptop
1981	Linguagens robóticas para facilitar gargalos de programação	
1982	John Warnock desenvolve o PostScript,	

	linguagem para definição de páginas e com Charles Geschke funda a Adobe Systems.	
1982	Sun Microsystems é formada	
1982	Microsoft, MS-DOS	
1983		Cray 2, 1 bilhão de FLOPs (floating point operations per second)
1983	At AT&T Bell Labs, Bjarne Stroustrup continua seu trabalho sobre o C++, uma extensão orientada a objeto para C	
1984		Apple, computador Macintosh
1984		IBM introduz o PC AT
1984	Programação Linear: algoritmo de Karmakar	
1984	Linguagem funcional Standard ML	
1984		CD-Rom, Sony e Philips
1985	Aldus apresenta o PageMaker, para o Macintosh: início da era da edição desktop	
1986	Início da popularização da palavra e conceito Case	
1986		Compaq, série Fortune 500; 1º Intel 80386-based PC
1986		HP, linha Spectrum, computadores com tecnologia RISC (reduced instruction set computers)
1986	Eiffel, linguagem orientada objeto	
1987		IBM PS/2
1987		Cray Research, Cray 2S, 40% mais rápido que Cray 2
1987		ETA Systems, ETA-10, supercomputadores
1987		Sun Microsystems, 1ª workstation baseada microprocessador RISC
1987	Watts Humphrey e William Sweet, Instituo de Engenharia de Software, EUA, publicam uma "estrutura de processos" que se torna um modelo para ajudar desenvolvimento de software confiável	
1987		Aldus, PageMaker para IBM PC e compatíveis
1987		Texas Instruments: 1º chip microprocessador baseado em IA
1988		Cray Research, Cray Y-MP, supercomputador de 20 milhõesde dólares

1988	IBM, sistema operacional MVS/ESA	
1988		Motorola, 88000, um microprocessador RISC
1988		O 1º supercomputador para aplicações gráficas, por Apollo, Ardent e Stellar
1988	AT&T anuncia plano de adquirir 20% da Sun Microsystems, que irá ajudar no desenvolvimento de nova versão para UNIX	
1988		Sun Microsystems 80386-based workstations
1988		Next workstation computer: 1º computador usando discos óticos apagáveis
1988	Internet network	Internet network
1988	Barry Boehm publica uma descrição do modelo espiral para desenvolvimento de software	
1989		DEC workstation usando computadores com tecnologia RISC
1989		Intel 80486 microprocessador e I860 RISC/coprocessador. Chips com mais de um milhão de transistores.
1989		Sun Microsystems, SPARCstation
1989		Cray se reestrutura em duas cias: Cray Research e Cray Computer Corp., esta liderada por Seymour Cray, o qual desenvolverá um supercomputador baseado em gálio-arsênio
1989		Mais de 100 milhões de computadores no mundo
1989		Pocket anuncia o 1º computador de pequeno porte com MS-DOS
1989		Grid introduz um laptop sensível ao toque, que reconhece escrita a mão
1989		Notebook com bateria: LTE e LTE/286 da Compaq
1989		DEC mainframe VAX 9000
1989		O 1º computador pessoal com tecnologia EISA
1989		O 1º computador baseado no chip 80486
1990	Berners-Lee escreve um protótipo inicial para a World Wide Web, que usa suas outras criações: URLs, HTML, e HTTP	
1990		Motorola microprocessador 68040

1990		IBM RISC Station 6000
1990		DEC VAX tolerante a falhas
1990		Cray Research, supercomputador, Y-MP2E
1990	Haskell, a última evolução das linguagens funcionais, para uso geral	
1990	Microsoft Windows 3.0	
1990	IBM PS/1	IBM PS/1
1990		IBM System 390, mainframe dos ano 90
1990		Microsoft Windows 3.0
1990		Apple, Classic, LC e IISI
1990		Intel i486 e iPSC/860 e Motorola 68040
1990		Sun Microsystems SPARCstation 2
1990		As primeiras estações SPARC compatíveis
1991		Advanced Micro Devices, microprocessador AMD 386 microprocessor, para competir com chip Intel 386
1991		Notebook PCs
1991		HP, série 700 RISC-based 9000
1991		Intel 486SX,
1991		NCR é assumida pela AT&T
1991		Sistema operacional Apple 7.0
1991		Microsoft DOS 5.0
1991		Borland compra Ashton-Tate
1991	SunSoft, subsidiária da Sun Microsystems anuncia o Solaris, sist.operacional UNIX para SPARC workstations e 386/486 PCs.	
1991		Wavetracer, Zephyr, computador paralelo com 8192 processadores
1992		IBM OS/2 2.0
1992		Microsoft Windows 3.1
1992		Sun Microsystems lança família SPARCstation
1992		Digital Equipment anuncia sua próxima geração de computadores com arquitetura baseada tecnologia RISC, o Alpha
1992		Microsoft Windows for Workgroup
1992		Intel Pentium
1992		Hewlett-Packard LaserJet 4, de alta resolução (600 x 600)

1992		Novell adquire UNIX Systems Laboratory
1993		Estudante e o staff do Centro de Aplicações Supercomputadores, Universidade de Illinoiscriam, criam uma interface gráfica para usuário para navegação na INTERNET chamada NCSA Mosaic
1993		CARMEL, robot de Univ. de Michigan, robótica auxiliada por computador
1993		Novell NetWare 4.0
1993		Motorola PowerPC, microprocessador
1993		Microsoft Windows NT
1993		Microsoft Plug and Play e Microsoft at Work (MAW)
1993		IBM workstation baseada em chip PowerPC
1993		IBM OS/2 para Windows
1993	Sun Microsystems licensia NextStep	
1994	Leonard Adleman, da Universidade de Southern California demonstra que o DNA pode ser usado como um medium para computadores	
1994	O 1º browser para o Netscape torna-se disponível e possibilta o rápido crescimento de surfistas WEB	
1994	Em abril, Jim Clark e Marc Andreesen fundam a Netscape Communications (originalmente Mosaic Communications)	
1994		Intel introduduz o 486DX4
1995	Linguagem de programação Java, apresentada em maio, possibilita desenvolvimento de aplicações com independência de plataforma	
1995		Em julho de 1995, pesquisadores da Universidade de Tokyo quebram a barreira do 1 teraflop com o processador 1.692-GRAPE-4(GRAvity PipE número 4), computador para aplicações especiais: simulações astrofísicas, especialmente os problemas gravitacionais
1995		Windows 95 é lançado dia 24 de agosto
1996		O Intel Pentium Pro é anunciado

1996		CRAYT3E-900, computador de uso geral com 1.8 teraflop
1997		Sistema StarMax 6000 da Motorola
1997		Lançamento do Windows NT 5.0
1997		Seagate Technology lança drive rígido para disco de 3,5” de 2,5 Gigabytes: o Seagate Medalist
1998		Versão 333 Mhz do processador Intel II. Nomeado Deschutes, este processador usa o novo processo industrial que o torna mais rapido, gerando menos calor que antes
1998		Nova versão do Windows 98
1999		Nova versão do kernel do Linux, a 2.2.0. O número de pessoas que usam Linux é calculado em mais de 10 milhões
1999		Nova versão do processador AMD Athlon: 750 Mhz
2000	MS Windows 2000	
2000		Novas versões dos processadores AMD Athlon e Intel Pentium III: 1 GHz
2000		Compaq iPAQ, para a plataforma DEC Itsy
2000	Mac OS X da Apple. Sistema operacional com interface gráfica baseada em Unix	
2000	Polêmica do bug do milénio (Y2K Bug)	
2001	Lançamento do Kernel do Linux 2.4	
2001		Lançamento do primeiro iPod da Apple
	MS Windows XP	
2002	Lançamento do Lindows Lindows alia-se à Microtel	
	Lançamento do Kernel do Linux 2.6	
2003	Microsoft Office 2003	
2003		2003 Motherboard Intel Canterwood
2003	Surge o vírus Worm Blaster	
2003		2003 Intel Prescott
2003	Multithreading	
2003		Comunicações WiFi
2004	Mozilla Firefox 1.0	
2004		nVidia releases GeForce 6800
2005	Windows Vista	
2005		Migração da plataforma Macintosh para processadores Intel

2006	Web 2.0	
2006		Apple lança o iPod Nano, o menor iPod com tela LCD e o iPod Video, com capacidade de armazenamento de até 80GB
2007		Maio: novo sistema Cray XT4 paraprevisão de tempo instalado no CSCS-Swiss National Supercomputing Centre: 2.6 GHz, 4.5 Tflops.

Nota: É de autoria individual a separação da evolução conceitual e tecnológica, observando-se que há casos onde é difícil se estabelecer uma distinção rígida entre idéia e equipamento. Saliente-se também que algumas das datas indicadas são conjeturais e, freqüentemente, controversas, e que, dada a multiplicidade das áreas e eventos dentro destas, muito possivelmente o quadro estará incompleto.

Anexo – O método axiomático e as ciências dedutivas*

O problema do método axiomático para Aristóteles surge da análise da estrutura de uma demonstração. Esta consta de três partes fundamentais: o que se quer demonstrar (ou seja, a conclusão), os axiomas (as premissas verdadeiras de que se parte) e um gênero cujas propriedades são objeto de demonstração (Analíticos II, A, VII, 75a, 39 – 75b,2).

Que toda demonstração tenha que partir de premissas que não podem ser objeto de demonstração e que qualquer definição deva se ater a uns poucos termos tomados como primitivos e não definíveis por seu lado, dentro do sistema, é provado por Aristóteles de maneira bastante clara. "Sustentamos, no entanto, que nem toda ciência é demonstrativa, mas sim que a do imediato não se constitui por demonstração. (É evidente que deve ser assim, pois se os antecedentes a partir dos quais se estabelece a demonstração devem ser conhecidos e se o processo demonstrativo deve terminar em proposições imediatas, é necessário que estas sejam indemonstráveis).

É evidente também que não é possível em absoluto demonstrar mediante um processo circular, visto que a demonstração parte de premissas prévias e mais conhecidas que a conclusão, e visto que uma coisa não poder ser ao mesmo tempo antecedente e conseqüente sob o mesmo aspecto, se bem possa ser prévia para nós enquanto é posterior em si mesma, como ocorre quando se conhece por indução... Àqueles que afirmam que é possível estabelecer demonstrações de caráter circular, pode-se objetar não somente pelo dito acima, mas também que se limitam a dizer que, se algo é, então é..." (Analíticos II, A, III, 72b, 18ss). Neste texto, junto a uma consideração de caráter gnoseológico (isto é, relativo a nossa maneira de conhecer) que é o reconhecimento de que as premissas imediatas devem ser evidentes, há uma pontualização de caráter claramente lógico-metodológico: a de que não pode haver demonstrações com um regresso ao infinito, e nem um processo circular, de modo que todo sistema dedutivo deve partir de axiomas.

Como um complemento, mais notável ainda do que apontar o método axiomático como o mais adequado para as ciências dedutivas é ter aplicado o método à própria lógica. Porque isso implica considerável dificuldade conceitual: axiomatizar uma teoria dedutiva significa essencialmente estabelecer certas premissas (os axiomas) e não admitir nela senão proposições (ou teoremas) deduzidas desses axiomas mediante o uso da lógica. Como poderá então pensar-se em axiomatizar a própria lógica? Porque para fazê-lo seria necessário adotar novos axiomas (os princípios lógicos) e depois fazer deduções a partir deles mediante o emprego da ... lógica! Ou seja, usar a lógica para a lógica. Não é possível aqui estudar isso, mas a grosso modo a solução – e foi o que Aristóteles fez para sua silogística – consiste em admitir que também na lógica é possível adotar certas estruturas consideradas como primitivas e, depois, 'extrair' outras delas mediante determinados procedimentos ou regras de transformação, que não têm por sua vez caráter de estruturas, mas sim o de operações verificáveis sobre estruturas.

Na matemática, uma axiomatização da teoria dos números havia sido oferecida desde 1888 pelo matemático alemão Richard Dedekind, em seu revolucionário tratado *Os*

---

* Baseado em **[Aga86]** e em artigo da Scientific American Brasil, edição Gênios da Ciência Matemática: A vanguarda matemática e os limites da razão

números: o que são e para que servem?, no qual buscava estabelecer “as propriedades da sucessão dos números naturais que sejam independentes, vale dizer, não se possam deduzir umas das outras, mas a partir das quais se possa construir todas as outras”. Assim, ele construiu a teoria dos números com base nos seguintes axiomas: 1) 1 é um número; 2) o sucessor de todo número é um número; 3) números distintos têm sucessores distintos; 4) 1 não é sucessor de nenhum número; 5) o conjunto dos números naturais é o menor conjunto S tal que 1 pertence a S e o sucessor de todo elemento de S também pertence a S. Esse último axioma, fundado no princípio de indução (ou de recorrência), havia permitido a Dedekind excluir da classe de modelos para sua teoria todas as estruturas que contivessem, para além dos números naturais, elementos “estrangeiros” (números que, depois, serão nomeados “não-standard”), e, assim, demonstrar a possibilidade de definir exatamente a estrutura dos números naturais.

No entanto, para construir seu sistema de axiomas, Dedekind havia utilizado de maneira informal a teoria dos conjuntos. Mais especificamente, ele colocara no mesmo nível objetos, expressões referidas a objetos e expressões referidas a outras expressões (ver o axioma 5): sua aritmética era de segunda ordem. Deve-se ao matemático italiano Guiseppe Peano a etapa seguinte, decisiva para a axiomatização da matemática. Em sua obra Arithmetices principia nova methodo exposita, publicada um ano depois dos trabalhos de Dedekind, Peano apresentou um sistema de axiomas para os números naturais que lembrava de maneira espantosa o sistema de Dedekind, apesar de concebido de modo independente. O matemático italiano, contudo, não construíra sua teoria dentro do contexto conjuntista, e introduziu uma notação (que, com uma ou outra modificação, tornou-se padrão) destinada a contornar certas ambigüidades inerentes à linguagem natural. Seu objetivo era captar, com o maior rigor possível, a natureza lógica do princípio de indução, ou seja, a lógica de segunda ordem.

Ver anexos *Dedução e Indução na Matemática* e *A Aritmética de Peano*.

Anexo – Dedução e Indução na Matemática*

* Baseado em **[Sho67], [Dan54]** e **[CO98]**

Uma das principais características da ciência Matemática, ao contrário das demais ciências, é o uso de provas em vez de observações. Um físico deve provar as leis físicas a partir de outras leis físicas, mas ele normalmente decide via observação: é a prova final de uma lei física. Um matemático eventualmente usa a observação. Por exemplo: ele pode medir os ângulos de vários triângulos e concluir que a soma dos ângulos é sempre 180°. Entretanto, ele somente aceitará isto como uma lei matemática quando estiver logicamente provado dentro de um sistema axiomático.

As leis que regem este tipo de raciocínio são antigas. Embora formuladas sistematicamente por Aristóteles, eram já conhecidas muito antes dele. Elas são uma espécie de espelho do intelecto humano: todo homem inteligente aplica de algum modo essas leis para alcançar os seus objetivos no dia a dia. Sabe que para raciocinar corretamente deve antes escolher algumas premissas sem ambigüidades, e então seguir passo a passo uma seqüência lógica de ações. Assim deverá chegar a uma única conseqüência, de acordo com o processo lógico seguido. Caso não chegue, irá provavelmente rever se aplicou corretamente as regras lógicas do processo, e se tudo foi aplicado corretamente, significará que há algo de errado em suas premissas.

Mas não é fácil estabelecer o conjunto de premissas para um determinado domínio de conhecimento: exige-se não só um juízo crítico apurado mas grande habilidade também, assim como é imperativo que cada premissa seja independente da outra e que todo o sistema esteja abrangendo a questão investigada. O campo da matemática que lida com tais problemas é chamado de *axiomática* e foi cultivado por homens do calibre de Peano, Russell e Hilbert. No anexo sobre *A concepção formalista da matemática* desenvolvem-se um pouco mais estas idéias.

Por ora basta saber que esse processo acima descrito é chamado de *dedutivo* e caracteriza o pensamento matemático. Ele encontrou sua completa realização na geometria, e por esta razão a estrutura lógica da geometria tornou-se modelo das ciências exatas.

De diferente natureza é outro método usado nas investigações científicas: a *indução*. Geralmente é descrito como o método que vai do particular para o geral. É o resultado de observações e experiências. Para se descobrir uma propriedade de uma certa classe de objetos, repetem-se testes tantas vezes quanto possível, sob circunstâncias semelhantes. Se uma determinada resposta tende a acontecer na maioria das vezes, tal tendência é aceita como uma propriedade daquela classe de objetos. Porém na matemática tal processo não pode ser utilizado, pois bastaria uma única resposta diferente das demais para negar uma determinada assertiva[*].

No entanto, algumas propriedades da aritmética, como a associativa, comutativa, etc., podem ser demonstradas por um método dedutivo chamado de *raciocínio por recorrência*, muitas vezes também denominado *indução matemática* ou *indução finita* ou ainda *indução completa*. Foi introduzido na teoria dos números pelo matemático italiano Giuseppe Peano, e desde então vem sendo vastamente aplicado na matemática e, em particular, na teoria dos conjuntos. Abaixo segue uma breve explicação desse procedimento, que está formalizado no anexo *A Aritmética de Peano*.

---

[*] Considere por exemplo a expressão $n^2 - n + 41$. Para n = 1,2,3,...40, gera-se em todos os casos um número primo. Seria um erro primário dentro da matemática pensar que tal expressão sempre gerará um número primo...

As propriedades principais do conjunto dos números naturais são:

1. Os números naturais podem ser gerados a partir do número natural 0 (zero) via a operação do sucessor: 'o sucessor de um número natural **n** é **n+1**, que é também natural'.
2. Quando uma determinada propriedade de números ocorre para um número natural e para o próximo número natural na geração, então a propriedade acontece para todos os números naturais.

Esta segunda propriedade é conhecida por princípio da indução finita e é assim enunciado: *Seja P uma propriedade de números naturais. Se* 0 *tem a propriedade P, e quando* n *tem a propriedade P,* n+1 *também tem a propriedade P, então todo natural tem a propriedade P.*

O princípio da indução é usado para demonstrar asserções, digamos P, sobre os números naturais e o procedimento de demonstração tem os seguintes passos:

(a) Base de indução: mostrar que 0 satisfaz a asserção P;

(b) Hipótese de indução: supor que o número natural k satisfaz a asserção P, e demonstrar que:

(c) Passo de indução: k+1 satisfaz a asserção P;

(d) Conclusão da indução: de (a), (b), (c) concluir que todo natural *n* satisfaz a asserção P.

Exemplo: Provar que $0 + 1 + 2 + 3 + ... + k = \frac{k(k+1)}{2}$

Base de indução: se k = 0, então $0 = \frac{0(0+1)}{2}$

Hipótese de indução: suponha válido para k = n, ou seja, $0 + 1 + ... + n = \frac{n(n+1)}{2}$

Passo de indução: para k = n + 1 tem-se

$$0 + 1 + 2 + ... + n + (n+1) = \frac{n(n+1)}{2} + (n+1) \quad \text{(por hipótese de indução)}$$
$$= \frac{n(n+1)}{2} + \frac{2(n+1)}{2}$$
$$= \frac{(n+1)(n+2)}{2}$$
$$= \frac{(n+1)((n+1)+1)}{2}$$

Conclusão de indução: a propriedade é válida

A possibilidade de usar indução finita para números naturais só é possível porque este conjunto é *indutivo*, isto é, existe um elemento inicial (no caso o 0) e todos os outros elementos são gerados pela aplicação da função sucessor, como segue abaixo:

Definição: O conjunto dos números naturais **N** é indutivo em A, onde A = {0}, e a função geradora é a operação de sucessor, que soma 1 a um número natural:

1. 0 é um número natural;
2. Se *a* é um número natural, então o sucessor de *a* é um número natural.
3. Os únicos números naturais são os objetos satisfazendo os itens 1 e 2 acima

Isto abre uma série de novas possibilidades de definições indutivas principalmente para conjuntos. É possível definir indutivamente conjuntos que possuam um conjunto de elementos iniciais e possuam um conjunto de funções geradoras.

A importância capital da indução matemática foi ressaltada, sobretudo, pelo grande pensador francês Henri Poincaré, no princípio do século XX. Poincaré fez ver que toda ciência matemática seria mera e estéril tautologia, redutível ao princípio da identidade, A = A, se o único modelo ali aplicado fosse o da *inferência silogístico-dedutiva*. Segundo Poincaré a prova por indução completa – que ele chamou de "dèmonstration par récurrence" – conteria uma "virtude criadora", capaz de possibilitar, de modo finito, a formulação de uma infinidade de juízos matemáticos.

Anexo - A aritmética de Peano*

Em 1889, o matemático italiano Giuseppe Peano resumiu as características estruturais dos números naturais em uma lista de axiomas enunciados em lógica simbólica. Esta última era uma linguagem de primeira ordem (ou seja, uma linguagem na qual aparecem somente predicados aplicados aos objetos da linguagem, mas não predicados aplicados aos predicados, nem proposições acerca de proposições), com identidade. A identidade (cujo símbolo é=´) fica definida por duas propriedades:

1) $a = a$; $a = b \rightarrow b = a$; $(a = b \wedge b = c) \rightarrow a = c$

➔ a igualdade é uma relação reflexiva, simétrica e transitiva

2) $a_1 = a_2 \rightarrow \varphi(a_1) = \varphi(a_2)$

➔ para dois objetos idênticos, sempre que um deles possuir uma propriedade φ, o outro também a possuirá

O conceito central da aritmética de Peano é o de ´sucessor´: todo número natural x tem um sucessor. Esse sucessor não pode ser escrito como x + 1, pois a adição ainda não foi definida. Peano indica então com s(x) ("sucessor de x") o número que segue a x e especifica que a função s está definida para todo número natural x. Ele formaliza assim uma propriedade importante dos números naturais ("pode-se contar sempre um a mais") e que, depois de especificada sua estrutura particular, servirá para estabelecer tacitamente que existem infinitos números.

As constantes da linguagem aritmética de Peano são as seguintes: 0 (o número zero), s (a função ´sucessor´), ´+´ e ´×´, as operações de adição e multiplicação. O significado destas constantes fica definido pelos seguintes axiomas:

1) $\forall x (\neg s(x) = 0)$

➔ 0 (zero) não é sucessor de número algum

2) $\forall x \forall y (s(x) = s(y) \rightarrow x = y)$

➔ números distintos têm sucessores distintos

3) $\forall \alpha (\alpha(0) \wedge \forall x (\alpha(x) \rightarrow \alpha(s(x))) \rightarrow \forall x \alpha(x))$

➔ esse é o princípio da indução matemática completa: se uma propriedade α é verdadeira para o zero e é verdadeira a frase "se α é verdadeira para x, então α é verdadeira para o sucessor de x", então α é verdadeira para qualquer x, isto é, para todo número natural.

4) $\forall x \forall y (x + 0 = x) \wedge x + s(y) = s(x+y)$

5) $\forall x \forall y (x \times 0 = 0) \wedge x \times s(y) = x \times y + x$

➔ estes dois últimos axiomas definem, por indução, a adição e multiplicação

* A maior parte do texto vem de artigo publicado na revista Scientific American Brasil, edição Gênios da Ciência Matemática: A vanguarda matemática e os limites da razão

Anexo - O Método das Diferenças

Se uma função, como por exemplo *f(x) = 2x + 3*, é avaliada para sucessivos valores de *x* e anotarmos as diferenças entre cada valor de f(x), encontraremos:

x	**f(x)**	**1ª diferença**
0	3	
1	5	2
2	7	2
3	9	2
4	11	2
5	13	2
6	15	2

Para se achar o valor de f(x) para x = 7, sem fazer nenhuma multiplicação, basta tomar a constante de diferença e somar a f(6) já calculado, portanto f(7) = 15 + 2 = 17.

Se a função fosse um pouco mais complexa, como por exemplo $f(x) = x^2 + 2x + 3$, então seria necessário obter as diferenças das diferenças (ou segundas diferenças) antes de se chegar ao valor constante. Por exemplo:

x	**f(x)**	**1ª diferença**	**2ª diferença**
0	3		
1	6	3	
2	11	5	2
3	18	7	2
4	27	9	2
5	38	11	2
6	51	13	2

Vamos agora encontrar o valor de f(x) para x = 7. O valor da constante de diferença é 2 que dever ser somado ao valor 13, encontrando-se assim o valor da coluna da 1ª diferença para x = 7. Logo, f(7) será 15 + 51 = 66.

Em geral, se a equação polinomial a ser avaliada tem um termo de aridade *n*, então será necessário ser tomada a n-ésima diferença antes de uma constante ser encontrada. Se é preciso avaliar uma equação polinomial para vários valores de *x*, tal como quando se está computando uma tabela, é mais fácil fazê-lo adicionando a diferença à diferença de cima, para então adicionar aquela diferença à de cima, e assim por diante até a o valor da função ser encontrado. Isto resulta em um procedimento no qual somente adições são exigidas se a própria função fosse ser avaliada para cada valor de *x*.

Embora todas as polinomiais tenham uma constante de diferença, funções logarítmicas e trigonométricas não têm, em geral, essa propriedade. Assim, para se poder usar o método das diferenças quando da produção de tabelas de tais funções, é necessário aproximar a função com uma polinomial e então avaliar essa polinomial.

Uma *Máquina de Diferenças* é simplesmente uma máquina com a capacidade de armazenar uma série de números e executar adições com eles. Os números irão representar os valores da função, sua primeira diferença, segunda, terceira, e assim por diante. Em função da máquina poder adicionar as diferenças inferiores às superiores e finalmente chegar ao valor da função, é possível gerar sucessivos valores da função. A máquina projetada por Babbage era capaz de trabalhar com polinomiais de grau seis **[Wil97] [Gol72]**.

Anexo - A concepção formalista da Matemática*

Antes das considerações mais técnicas sobre o formalismo na Matemática, é oportuno fazer alguns comentários sobre o conceito 'forma' (com "o" aberto), do ponto de vista da Lógica, mais especificamente de um dos seus ramos, que é a Lógica Matemática ou também Lógica Simbólica.

Ainda que o sentido mais intuitivo do termo forma relacione-se com a configuração externa dos objetos materiais, também é costume, na linguagem ordinária, falar de forma em um sentido mais amplo, como por exemplo, quando se comenta que uma composição poética está em *forma* de soneto, ou que uma composição musical está em *forma* de sonata. O que se pensa nesse momento é nas propriedades *estruturais* que são observáveis, sem ter em conta o significado dos versos ou dos motivos que inspiraram a música. Da mesma maneira usa-se o termo estrutura não só para indicar a constituição de um corpo sólido, mas também se referindo à estrutura de uma sociedade, de um discurso, e assim por diante.

Do mesmo modo pode-se pensar em *estruturas lógicas* ou *formas lógicas*, e, dentro da ciência Lógica, tais expressões representam um aspecto que se reveste de capital importância: o aspecto formal. A lógica formal é um tipo de investigação sobre a linguagem que simplesmente analisa as estruturas desta, prescindindo de conteúdos concretos que posteriormente sejam dados a estas estruturas (gerando proposições concretas de um discurso falado ou escrito). Por exemplo: "todos os A são B, todos os B são C, e portanto todos os A são C". De tal tipo de estrutura surgem argumentos <u>válidos</u>, quaisquer que sejam os termos usados para substituir A, B ou C (embora a conclusão possa ser falsa se uma das premissas for falsa†).

A forma lógica diz respeito ao conteúdo dos nexos que organizam uma demonstração (um raciocínio dedutivo), prescindindo-se dos conteúdos semânticos do discurso. A lógica, nesse caso, somente se ocupa do problema do desenvolvimento dessa demonstração. O fato de prescindir dos conteúdos tem, entre outras conseqüências, a de que é possível utilizar-se de estruturas dedutivas mediante símbolos, e isso permite uma exatidão da análise estrutural que seria muito mais difícil de conseguir sem o auxílio do simbolismo‡ **[Aga86]**.

A computação é sobretudo a ciência do formal. Os computadores seguem fielmente regras e não admitem exceções. O programa não funciona quando se troca um '0' (zero) por 'O' (letra ó), engano de natureza apenas formal. A Matemática também é formal, a mais formal de todas as ciências (pois todos os seus resultados são baseados em regras e

* Este resumo baseia-se nas exposições feitas em **[NN56]** e **[Cos77]**.

† O ponto de vista da correção e da verdade de um raciocínio é distinto dentro da Lógica, embora voltem a unir-se principalmente considerando-se que as regras lógicas permitem obter de premissas verdadeiras somente conclusões verdadeiras.

‡ "De qualquer maneira, não é preciso reduzir o horizonte conceitual da lógica simbólica a este simples 'aspecto instrumental' que, apesar de ser o mais facilmente compreendido de início, tem o risco de fazer perder de vista a verdadeira natureza dos problemas abordados pela moderna logística. De fato, os desenvolvimentos desta última ultrapassam amplamente a tarefa – no fundo bastante modesta – de proporcionar instrumentos mais precisos para o estudo da dedução. Com efeito, a Lógica já é uma verdadeira ciência por si mesma, que é estudada e desenvolvida com o mesmo interesse puramente especulativo que move as investigações das matemáticas puras ou da álgebra abstrata" **[Aga86]**.

apresentados por fórmulas), e sua linguagem formal é utilizada por todas as outras. No entanto os formalistas, escola fundada pelo prof. David Hilbert, são apenas um dos grupos dentro da Matemática e seus resultados foram e são fortemente questionados*.

Hilbert, diferentemente dos matemáticos da escola logicista, não tinha pretensões de reduzir a matemática à lógica, mas fundamentar conjuntamente ambas. Ele e os outros seguidores da escola formalista viam na matemática a ciência da estrutura dos objetos. Os números são as propriedades estruturais mais simples desses objetos e por sua vez constituem-se também em objetos, com novas propriedades. O matemático pode estudar as propriedades dos objetos somente por meio de um sistema apropriado de símbolos, reconhecendo e relevando os aspectos destituídos de importância dos sinais que utiliza. Uma vez que se possua um sistema de sinais adequados, não é mais necessário se preocupar com seus significados: os próprios símbolos possuem as propriedades estruturais que interessam. Aqui devemos atentar para o fato de que a *formalização* não deve ser confundida com este aspecto não essencial que é a *simbolização.* O matemático deve apenas investigar, segundo os formalistas, as propriedades estruturais dos símbolos, e portanto dos objetos, independentemente de seus significados. Assim como na geometria ou na álgebra, para simplificar e uniformizar determinadas questões, são introduzidos conceitos não reais – ponto do infinito, números ideais, etc. – que são apenas convenções lingüísticas, também se justifica a introdução, na matemática, de conceitos e princípios sem conteúdo intuitivo. Desse modo, as leis da lógica clássica permanecem válidas.

Ponto chave na metamatemática† de Hilbert é que o sistema estudado não encerre contradição, isto é que não se possa provar uma proposição e ao mesmo tempo a sua negação. Ele procurou estabelecer um método para se construir provas absolutas de consistência (ausência de contradição) dos sistemas, sem dar por suposta a consistência de algum outro sistema.

O primeiro passo é a completa formalização de um sistema dedutivo. Isto implica 'tirar' todo significado das *expressões* existentes dentro do sistema, isto é, devem ser consideradas puros sinais 'vazios'. Expressão é o nome que se dá às 'palavras' do sistema, que por sua vez são compostas de símbolos abstratos, também chamados 'alfabeto' do sistema. A forma como se devem combinar essas expressões deve estar plasmada em um conjunto de *regras de formação* e *regras de inferência* enunciadas com toda precisão, que especificam como uma expressão pode ser formada ou transformada em outra. A finalidade desse procedimento é construir um 'cálculo' que não oculte nada e que somente contenha o que expressamente se tenha colocado nele. Em um sistema formal um número finito de expressões é tomado como sendo o conjunto de *axiomas* do sistema. A idéia de prova num sistema formal consiste em começar com um dos axiomas e aplicar uma seqüência finita de transformações, convertendo o axioma em uma sucessão de novas expressões, onde cada uma delas ou é um dos axiomas do sistema ou é derivada deles pela aplicação das regras de formação. A totalidade dos teoremas constitui o que pode ser provado no sistema. Os axiomas e os teoremas de um

* Na verdade, a maioria dos matemáticos desenvolve seus resultados dentro de um espírito mais informal, intuitivo, mais geométrico do que algébrico, e quando algébrico, não muito formal.

† Quando a Matemática fala da Matemática

sistema completamente formalizado são portanto sucessões de comprimento finito de símbolos sem significado.

Especificando um pouco melhor (baseado em **[CO98]**):

Definição 1: *Um sistema formal / é uma tupla* $<\Sigma, L, A, R>$, *onde:*

1. Σ *é um alfabeto;*
2. L *é um conjunto recursivo*[*] *em* Σ, *chamado de linguagem do sistema formal;*
3. A *é um subconjunto recursivo de* L, *chamado de* ***Axiomas****;*
4. R *é um conjunto recursivo de relações em* L.

Exemplo: seja um sistema formal, onde o alfabeto, as palavras, os axiomas e as relações estejam definidas abaixo:

$$\Sigma = \{+,*\},\ L = \{\Sigma^*\},\ A = \{+,*\},\ R = \{r_1, r_2\},\ \text{onde:}$$

$$r_1 = \{<x+, x*> \mid x \in \Sigma^*\}$$

$$r_2 = \{\{<x+*, x*+> \mid x \in \Sigma^*\}\ U$$
$$\{<x+**, x*++> \mid x \in \Sigma^*\}\ U$$
$$\{<x*, x++> \mid x \in \Sigma^*\}\}$$

As relações r_1 *e* r_2 *são binárias, e seus pares ordenados possuem uma lei de formação bem definida.*

Definição 2: *seja* $/ = <\Sigma, L, A, R>$ *um sistema formal e seja o conjunto* $\Gamma \subseteq L$. *Uma dedução ou derivação de* α *a partir de* Γ *em / é uma seqüência* $\alpha_1, \alpha_2, \ldots \alpha_n$ *de palavras de* L, *tal que:*

1. α_n *é* α*; e*
2. *para todo* j, $1 \leq j < n$,
   (a) *ou* $\alpha_j \in \Gamma\, U\, A$
   (b) *ou existem* $\alpha_{j1}, \alpha_{j2}, \ldots \alpha_{jk}$, $j_i \in \{1, \ldots, j-1\}$, $1 \leq i < k$, *tais que* $<\alpha_{j1}, \alpha_{j2}, \ldots \alpha_{jk}> \in r$, *com* $r \in R$

Se existir uma dedução de α *a partir de* Γ *dizemos que* α *é dedutível ou derivável, a partir de* Γ *em /. Isto é denotado por* Γ / α.

Exemplo: no sistema formal do exemplo acima, uma dedução de *+ é:

$+ \quad (\in A)$
$* \quad (<+,*> \in r_1)$
$++ \quad (<*,++> \in r_2)$
$+* \quad (<++,+*> \in r_1)$
$*+ \quad (<+*,*+> \in r_2)$

---

[*] Não aprofundando muito na questão, que entra na árdua teoria da recursão: um conjunto A contido em X é recursivo se e somente se há um procedimento pelo qual, dado um x pertencente a X, pode-se computar sobre x, pertença x a A ou não. Exemplo: o conjunto dos primos em N (o procedimento é o crivo de Eratóstenes)**[Sho71]**.

portanto a seqüência *,++,+*,*+ *é uma dedução de* *+ *onde* $\Gamma = \{\}$; *assim* {}/*+.

Uma axiomática formalizada converte-se, em resumo, em uma espécie de jogo grafo-mecânico, efetuado com símbolos destituídos de significado e regulado por meio de regras determinadas. E isso tem uma valiosa finalidade: revelar com clareza a estrutura e a função, similarmente ao manual esquemático e de funcionamento de uma máquina. Quando um sistema está formalizado, tornam-se visíveis as relações lógicas existentes entre as proposições matemáticas, como se combinam, como permanecem unidas, etc.

Uma página inteira preenchida com os sinais 'sem significado' não afirma nada: é simplesmente um desenho abstrato de um mosaico que possui determinada estrutura. No entanto é perfeitamente possível descrever as configurações de um sistema assim especificado e formular declarações acerca das configurações e das suas diversas relações mútuas. Hilbert observou que tais declarações pertencem à *metamatemática,* isto é, declarações a respeito dos símbolos e expressões existentes dentro de um sistema matemático formalizado.

Para cada sistema formalizado procura-se provar sua consistência, evidenciando-se que jamais se poderá chegar a arranjos simbólicos contraditórios. Os métodos utilizados foram denominados por Hilbert de métodos *finitísticos*: procedimentos elementares e intuitivos de tipo combinatório, utilizados para manipular um número finito de objetos e funções bem determinadas[*]. A quantidade de axiomas e regras do sistema tinha de ser construtível com um número finito de passos e que os enunciados passíveis de prova tinham de ser provados com um número finito de passos.

Como um sistema formal sintático se relaciona com um mundo de objetos matemáticos aos quais estão associados significados? Esta relação se dá através da noção de *interpretação*[†]. Desta forma todos os teoremas do sistema formal podem ser interpretados como enunciados verdadeiros acerca desses objetos matemáticos. O sonho de Hilbert era encontrar um sistema formal no qual todas as verdades matemáticas fossem traduzíveis, mediante algum tipo de interpretação, para teoremas e vice-versa. Tal sistema é denominado *completo*. O teorema de Gödel veio a destruir esse sonho.

---

[*] Como se pode notar Hilbert utiliza-se da intuição, mas não como os intuicionistas no sentido de estabelecer as propriedades de determinados entes matemáticos, mas referindo-se unicamente à efetuação de operações muito simples, tão seguras e elementares a ponto de serem aceitas na base de qualquer pesquisa teórica.

[†] Uma interpretação é a descoberta de um *isomorfismo* entre duas estruturas: no caso ela confere significado aos objetos e entidades matemáticas, tais como linha, ponto, símbolos abstratos, etc.

Anexo - O problema da decisão na Matemática*

Os casos mais conhecidos e elementares do problema da decisão pertencem à aritmética. Por exemplo, dados dois inteiros *a* e *b,* como descobrir se *a* é exatamente divisível por *b*? Para responder a tal questão não é necessária uma especial intuição, já que existe um procedimento de cálculo, puramente mecânico (no caso executar a conhecida operação matemática da divisão aritmética de *a* por *b*), que permite chegar ao término da operação após um número finito de passos, obtendo um quociente e um resto, depois do qual só são possíveis dois casos: ou *o resto é zero* e então se diz que *a é divisível por b*, ou o *resto é diferente de zero*, e então *a não é exatamente divisível por b.* Existem ainda casos que envolvem um procedimento decisório mais simples como o que responde à pergunta se um número é divisível por 2. Não é, nesse caso, necessário executar a operação de divisão, bastando ver se a expressão decimal de *a* termina em 0, 2, 4, 6 ou 8. Da mesma forma existe um procedimento mais simples para ver se *a* é divisível por 3, por 9 e alguns outros números.

Nesses casos, o problema da decisão relativo às perguntas simples colocadas é solucionável, pois existe um procedimento decisório, que de maneira aritmética e finita, oferece a possibilidade de responder afirmativamente ou negativamente às perguntas citadas. No entanto, existem na aritmética elementar perguntas também simples, para as quais não existe ainda um procedimento decisório, como por exemplo, se os pares de números primos que se sucedem imediatamente na série dos números ímpares são finitos ou infinitos em número. Quer dizer, se pares como 11-13, 17-19, 41-43, ... são finitos ou infinitos. Como não há um procedimento de cálculo que ofereça uma, por assim dizer, 'lei de geração' para o conjunto dos números primos, não se está em condição de responder a essa pergunta†.

* Conforme **[Aga86]**

† Ao que parece, conforme vão se gerando maiores números primos, esses pares vão escasseando. Mas se deixarão de aparecer não se sabe.

Anexo - O Teorema da Incompletude de Gödel*

A numeração de Gödel

Para conferir à metamatemática o rigor necessário, Gödel tenta formalizá-la por meio de uma teoria com duas características essenciais: por um lado, uma teoria poderosa o suficiente para expressar a metateoria sintática; por outro, uma teoria que pudesse ser construída em um número finito de etapas, de acordo com a exigência finitista. A aritmética de Peano, que apresenta essas duas propriedades, será essa teoria. O desafio, assim, está em traduzir os enunciados da metalinguagem da aritmética de Peano na linguagem-objeto da aritmética. Os objetos da linguagem-objeto são números; os da metalinguagem são afirmações acerca dos números: Gödel precisa encontrar um modo de expressar tais afirmações por meio dos próprios números.

Ele procede assim: a cada símbolo da aritmética de Peano, ele atribui de maneira unívoca um número, chamado seu número de Gödel. A partir daí, é possível atribuir, também de maneira unívoca, um número de Gödel para todas as outras expressões da aritmética de Peano, bem como para todas as suas fórmulas e todas as suas seqüências finitas de fórmulas. A eficácia desse método fica garantida pelo fato de que essa aritmetização da linguagem (como o processo é designado) acontece em duas etapas: toda função da aritmética pode ser deduzida com auxílio de certas funções básicas chamadas de *funções recursivas primitivas*†, as quais são sempre calculáveis por construção.

Por meio de uma tabela de correspondência, Gödel atribui a cada símbolo da aritmética de Peano um número ímpar: “0” é traduzido por 1; o sucessor s por 3; a negação por 5; o símbolo $\vee$ por 7; $\forall$ por 9; “(“ por 11; “)” por 13; e as variáveis de tipo *n* por números da forma p^n, em que p é um número primo superior a 13. Uma fórmula da aritmética de Peano, que é uma seqüência desses símbolos, é levada, portanto, em uma correspondente seqüência de números ímpares: $n_1, n_2, ..., n_k$ Essa seqüência, por sua vez, é transformada em um número único *m*, por meio da seguinte instrução (que é uma função recursiva primitiva):

$$m = 2^{n1} \times 3^{n2} \times ... \times p_k^{nk}$$

Em que p_k é o k-ésimo número primo (em outras palavras, o segundo membro é a decomposição de *m* em fatores primos). O número *m* é o número de Gödel da fórmula. Assim, $n_1, n_2, ... n_k$ são os números de Gödel dos símbolos de uma fórmula da aritmética de Peano, e *m* é o número de Gödel dessa fórmula. Esse processo pode ser repetido para uma seqüência de números de Gödel associados a uma seqüência de fórmulas da aritmética de

---

* A maior parte do texto vem de artigo publicado na revista Scientific American Brasil, edição Gênios da Ciência Matemática: A vanguarda matemática e os limites da razão e também tem como base **[Cas97]**

† Uma função é calculável quando existe um algoritmo que, para todo argumento dado, fornece o valor da função em um número finito de etapas. Para enfrentar a infinidade de algoritmos imagináveis, pode-se considerar uma classe das funções calculáveis: as funções recursivas primitivas. Elas sempre tomam números naturais por argumento, e são definidas da seguinte maneira: certas funções simples – denominadas ´funções base´ - são declaradas, inicialmente, como recursivas primitivas. Chamam-se recursivas primitivas depois todas as funções que possam ser construídas, de acordo com certas regras, a partir das funções que já se saiba serem recursivas primitivas. A demonstração do teorema da incompletude de Gödel baseia-se em uma série de funções recursivas primitivas deduzidas umas a partir das outras.

Peano, o que resulta em um número de Gödel para essa seqüência de fórmulas. Essa codificação para as seqüências de fórmulas é importante, pois as demonstrações nada mais são do que seqüências finitas de fórmulas, em que cada uma ou é um axioma, ou decorre das fórmulas precedentes.

Os teoremas da teoria dos números asseguram que a enumeração de Gödel é unívoca, vale dizer, que a cada símbolo, fórmula ou seqüência de fórmulas corresponde um único número de Gödel, que lhe é exclusivo. Em outras palavras, é possível saber, para cada número natural, a partir da unicidade de sua decomposição em fatores primos, se esse número é um número de Gödel e, nesse caso, de qual elemento (seja símbolo, fórmula ou seqüência de fórmula) ele é símbolo.

O método de Gödel (a aritmetização), apresentado aqui para as fórmulas e demonstrações do sistema formal da aritmética de Peano, pode ser reproduzido de maneira semelhante para não importa qual linguagem (notadamente linguagens de programação). Atribuem-se números aos elementos básicos da linguagem (letras, palavras, caracteres especiais) e formam-se novos números a partir das seqüências desses números, de acordo com a instrução recursiva oferecida acima. Graças à aritmetização, os conceitos metalingüísticos da sintaxe da aritmética de Peano aparecem agora traduzidos como propriedades, funções ou relações entre números. Por exemplo, o conceito "é uma fórmula" corresponde à propriedade "é um número cujos expoentes da decomposição em fatores primos são todos ímpares". Analogamente (embora com um grau bem maior de dificuldade), o conceito metateórico "x é uma fórmula demonstrável", indicado pelo símbolo *Dem* (de demonstrável), pode ser expresso com auxílio de relações aritméticas.

O Teorema de Incompletude

Após estabelecer, em quatro teoremas, um método para a construção de funções recursivas, Gödel enuncia uma seqüência de 45 propriedades e funções, "cada uma das quais é definida com base nas precedentes por meio dos procedimentos dados nos teoremas I a IV". A função número 45 é uma função *D* de duas variáveis: y*D*x significa "a seqüência de fórmulas de número de Gödel y é uma demonstração para a fórmula de número de Gödel x". Sob o número 46, ele define enfim a demonstrabilidade:

Dem(x) ≡ (∃y)yDx,

e acrescenta entre parênteses: *"Dem(x)* é o único conceito, entre todos os definidos, de 1 a 46, a respeito do qual não podemos afirmar que seja recursivo". A fórmula 46 deve ser interpretada assim: "A fórmula x é demonstrável se, e somente se, existe uma seqüência q de fórmulas que demonstra x". Existe, portanto, uma fórmula para a frase "x é uma fórmula demonstrável", bem como para sua negação.

Gödel mostra, além disso, que a função de substituição *subst,* que permite substituir uma variável por um valor numérico dentro de uma fórmula, é uma função recursiva primitiva. *Subst* é de importância capital para a demonstração de Gódel, pois fornece a chave para a auto-referência: permite inserir o número de Gödel da fórmula "a proposição x não é demonstrável" no lugar da própria variável x.

Mais especificamente, chamemos de F(x) a fórmula "a proposição de número de Gödel x não é demonstrável"; seja *f* seu número de Gödel. A função de substituição permite substituir, na própria fórmula F(x), a variável x por esse número de Gödel *f* Obtemos assim a fórmula F(*f*): `*A* proposição de número de Gödel *f* não é demonstrável", ou seja, "a proposição a proposição de número de Gödel x não é demonstrável não é demonstrável". Isso significa que, em um sistema não-contraditório de axiomas, a fórmula F(*f*) não é formalmente demonstrável. Em uma etapa seguinte, Gödel demonstra que a fórmula F(*f*), ainda que formalmente não-demonstrável, é uma proposição aritmética verdadeira para todos os números inteiros. Como proposição verdadeira, sua negação (que é falsa) também não pode ser formalmente demonstrada no sistema. Segue daí que a proposição F(*f*) é indecidível - que não pode ser deduzida. A fórmula F(*f*), portanto, é ao mesmo tempo verdadeira e formalmente indecidível.

Uma vez atingido esse resultado, Gödel observa: "O método de demonstração que foi exposto pode-se aplicar a todo sistema formal que, em primeiro lugar, interpretado como sistema de conceitos e proposições, ofereça recursos expressivos suficientes para definir os conceitos que aparecem no raciocínio precedente (em particular, o conceito fórmula demonstrável), e no qual, em segundo lugar, toda fórmula demonstrável seja verdadeira na interpretação considerada". Gödel chega assim a seu teorema de incompletude, que pode ser enunciado da seguinte maneira: "Toda teoria axiomatizada suficientemente poderosa para expressar a aritmética é incompleta".

Em seu breve livro sobre Gödel, Jaako Hintikka sublinha que, ao contrário do que levam a crer diversas vulgarizações desse resultado revolucionário, o primeiro teorema de incompletude de Gödel não demonstra que "existem na aritmética (ou em outro sistema) proposições verdadeiras, mas absolutamente indemonstráveis. Ele mostra, antes, que todas as proposições verdadeiras da aritmética não podem ser demonstradas por meio de um único sistema formal dado".

Para Gödel, sistema formal e procedimento determinista e mecânico andam juntos. Assim, convencido de que as funções recursivas primitivas não dão conta do conceito de procedimento mecânico de maneira satisfatória, ele tenta desenvolver uma versão generalizada da recursividade. Em 1936, o lógico americano Alonzo Church (1903-1995) demonstrará, a partir dessas pesquisas, a indecidibilidade da lógica de predicados de primeira ordem: é impossível obter, para a lógica de predicados de primeira ordem, um procedimento geral de cálculo capaz de determinar, para toda fórmula, se ela é ou não válida. Seu teorema, conhecido atualmente como teorema de Church, responde assim negativamente ao problema de decidibilidade proposto por Hilbert em seu programa: "Um problema matemático dado", pensava Hilbert, "deve admitir, obrigatoriamente, uma solução exata, seja sob a forma de uma resposta direta a uma questão colocada, seja pela demonstração de seu caráter insolúvel e do fracasso inevitável de toda tentativa nesse sentido". Baseados também nas pesquisas de Gödel, os trabalhos de Alan Turing (1912-1954) constituirão o corolário do teorema de Church em informática teórica.

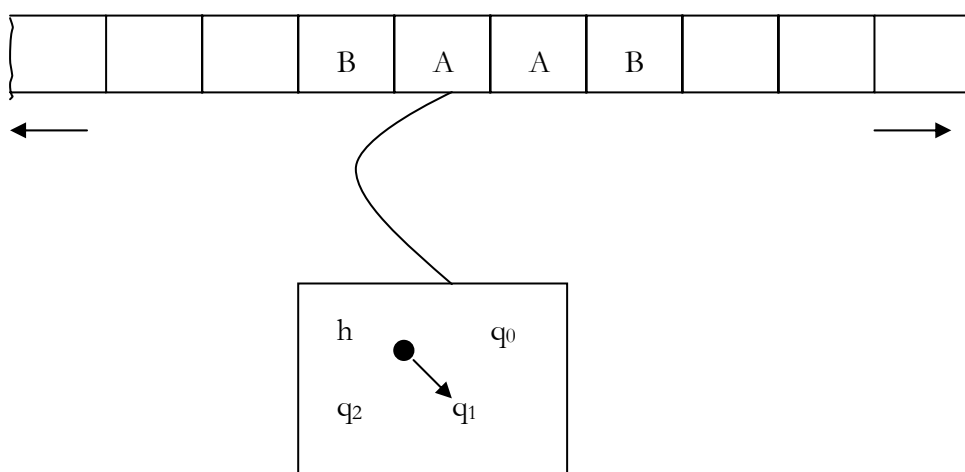
De muitos modos o trabalho de Gödel aconteceu também em outras áreas. Apenas quatro anos antes dele publicar o seu trabalho, o físico alemão Werner Heisenberg descobriu o princípio da incerteza. Assim como existe um limite fundamental nos teoremas que os

matemáticos poderiam provar, Heisenberg mostrou que havia um limite fundamental nas propriedades que os físicos poderiam medir. Por exemplo, se eles queriam medir a posição exata de um objeto, então só poderiam medir a velocidade do mesmo com uma precisão muito pobre. Isto acontece porque para medir a posição do objeto seria preciso iluminá-lo com fótons de luz, mas, para determinar a localização exata, os fótons precisariam ter uma energia enorme. Contudo, se o objeto está sendo bombardeado com fótons de alta energia, sua própria velocidade será afetada e se tornará inerentemente incerta. Portanto, ao exigir o conhecimento da posição de um objeto, os físicos teriam de desistir do conhecimento de sua velocidade.

O princípio da incerteza de Heisenberg só se revela nas escalas atômicas, quando medidas de alta precisão se tornam críticas. Logo, uma boa parte da física pode ser realizada sem problemas enquanto os físicos quânticos se preocupam com as questões profundas sobre os limites do conhecimento. O mesmo acontecia no mundo da matemática. Enquanto os lógicos se ocupavam do debate altamente abstrato sobre a indecidibilidade, o resto da comunidade continuava seu trabalho sem preocupação. Gödel tinha provado que existiam algumas afirmações – até infinitas – que não poderiam ser provadas, mas restava uma outra quantidade que podiam ser provadas e sua descoberta não invalidava nada que tivesse sido demostrado no passado. Além disso, muitos matemáticos acreditavam que as declarações de indecidibilidade de Gödel seriam encontradas nas regiões mais extremas e obscuras da matemática e, portanto, talvez nunca tivessem de ser enfrentadas. Afinal, Gödel só dissera que essas afirmações indecidíveis existam; ele não pudera apontar uma. Então, em 1963, o pesadelo teórico de Gödel se tornou uma realidade viva.

Paul Cohen, um matemático de 29 anos, da Universidade de Stanford, desenvolvera uma técnica para testar se uma afirmação particular é indecidível. A técnica só funcionava para certos casos muito especiais, mas, de qualquer forma, ele foi a primeira pessoa a descobrir que havia questões de fato que eram indecidíveis. Tendo feito sua descoberta, Cohen imediatamente voou para Princeton, com a demonstração na mão, de modo que fosse verificada pelo próprio Gödel. Dois dias depois de receber o trabalho, Gödel deu a Cohen sua aprovação. E o que era particularmente dramático é que algumas dessas questões indecidíveis estavam no centro da matemática. Ironicamente Cohen provara que uma das perguntas que David Hilbert colocara entre os 23 problemas mais importantes da matemática, a *hipótese do continuum*, era indecidível.

Anexo - Máquinas de Turing

O processo computacional foi graficamente mostrado no artigo de Turing, *On Computable Numbers with an aplication to the Entscheidungsproblem*, quando ele pediu ao leitor que considerasse um dispositivo que pudesse ler e escrever símbolos em uma fita que estava dividida em quadrados. Uma cabeça de leitura/gravação se moveria em qualquer direção ao longo da fita, um quadrado por vez, e uma unidade de controle poderia interpretar uma lista de instruções simples sobre leitura e gravação de símbolos nos quadrados, movendo-se ou não para a direita ou esquerda. O quadrado que é "lido" em cada etapa é conhecido como "quadrado ativo". A regra que está sendo executada determina o que se convencionou chamar 'estado' da máquina. A fita é potencialmente infinita (ver figura).

Imagine os símbolos "A" e "#"(branco). Suponha que o dispositivo possa limpar qualquer um deles quando os lê em um quadrado ativo e trocá-lo por outro (i.é., apagar "A" e substituir por "#" e vice-versa). Lembre-se que o dispositivo pode mover a cabeça de leitura/gravação para a direita ou esquerda, de acordo com instruções interpretadas pela unidade de controle. As instruções podem limpar um símbolo, escrevê-lo ou deixá-lo como está, de acordo com o símbolo lido.

Qualquer tipo de "jogo" pode ser elaborado usando estas regras, não tendo necessariamente algum significado. Uma das primeiras coisa que Alan Turing demonstrou foi que alguns "jogos" construídos sob essas regras podem ser sofisticados, em contraste com a simplicidade destas operações primitivas.

Dado um quadrado que seja uma posição inicial de uma seção da fita preenchida por quaisquer caracteres ou brancos, o dispositivo executa ações especificadas por uma lista de regras, seguindo-as uma por vez até chegar àquela que force sua parada (se não há uma instrução explícita na tabela para uma determinada configuração da fita, então não há nada que a máquina possa fazer quando alcança aquela configuração, encerrando a execução portanto).

Cada instrução – ou regra – estabelece uma ação a ser executada se houver determinado símbolo no quadrado ativo no tempo em que é lido. No nosso caso vamos estabelecer 4 diferentes tipos de regra:

(a) Substituir #(branco) por símbolo
(b) Substituir símbolo por branco(#)
(c) Ir um quadrado para a direita
(d) Ir um quadrado para a esquerda

Um exemplo de instrução seria: “Se houver um A no quadrado ativo, substitua-o por #”. Esta instrução faz a máquina executar a segunda ação da lista acima. Para se elaborar um "jogo" é preciso fazer uma lista que especifique o número da regra que se deve observar no momento atual, e, de alguma forma, qual será a próxima. Cada regra desta lista será composta pela seguinte seqüência: o número da regra – estado da máquina –, um caracter ou branco(#) para comparação, próximo estado e ação (novo símbolo que irá para o quadrado ou movimentar para direita(>)/esquerda(<) a cabeça de leitura/gravação).

Segue abaixo uma lista de regras – código e descrição – que dirão a uma máquina de Turing como desenvolver um determinado "jogo":

1 A 2 #

Estado 1: se há um A no quadrado ativo, substitua-o por # e vá para estado 2;

2 # 3 >

Estado 2: se há um # no quadrado ativo, vá para estado 3 e ande um quadrado a direita;

3 A 3 >

Estado 3: se há um A no quadrado ativo, vá para estado 3 e ande um quadrado a direita;

3 # 4 >

Estado 3: se há um # no quadrado ativo, vá para estado 4 e ande um quadrado a direita;

4 A 4 >

Estado 4: se há um A no quadrado ativo, vá para estado 4 e ande um quadrado a direita;

4 # 5 A

Estado 4: se há um # no quadrado ativo, substitua-o por A vá para estado 5;

5 A 5 >

Estado 5: se há um A no quadrado ativo, vá para estado 5 e ande um quadrado a direita;

5 # 6 A

Estado 5: se há um # no quadrado ativo, substitua-o por A vá para estado 6;

6 A 6 <

Estado 6: se há um A no quadrado ativo, vá para estado 6 e ande um quadrado a esquerda;

6 # 7 <

Estado 6: se há um # no quadrado ativo, vá para estado 7 e ande um quadrado a esquerda;

7 A 8 <

Estado 7: se há um A no quadrado ativo, vá para estado 8 e ande um quadrado a esquerda;

8 I 8 <

Estado 8: se há um A no quadrado ativo, vá para estado 8 e ande um quadrado a esquerda;

8 # 1 >

Estado 8: se há um # no quadrado ativo, vá para estado 1 e ande um quadrado a direita;

Note que se houver um # no quadrado ativo quando os estados forem 1 ou 7, ou se há um A no quadrado ativo quando o estado da máquina é 2, ela pára, pois não saberia o que fazer.

O jogo neste caso é duplicar uma seqüência de A's que estejam na fita. Se a fita contiver AAAA, no final conterá AAAAAAAA. Para se jogar (em termos mais técnicos diríamos executar o programa descrito na lista de regras) é necessário especificar uma configuração inicial na fita, qual o quadrado inicial ativo e o estado inicial da máquina. Quando a máquina começar a executar, ela, a partir do estado inicial e do quadrado ativo seguirá a seqüência (lógica) de regras que darão o produto final.

Provavelmente poderá ser comentado que parecem coisas mecânicas demais, mas era precisamente isto o que Turing estava procurando. A lista de instruções pode ser seguida por um dispositivo mecânico.

Em sua essência, toda máquina de Turing move-se ou move símbolos, de uma posição para outra em uma fita, da mesma maneira que no exemplo dado acima. Nos dias de hoje estes símbolos podem ser impulsos eletrônicos em um microcircuito e a fita uma série de endereços de memória em um chip, mas a idéia é a mesma. Turing provou que sua hipotética máquina é uma versão automatizada de um sistema formal especificado por uma combinação inicial de símbolos (o conjunto de "A"s na fita no início do processo) e as regras (aquelas instruções escritas). Os movimentos são mudanças de 'estado' da máquina que correspondem a específicos passos de computação.

Os modernos computadores e inclusive este PC no qual está sendo escrito este livro parecem ser bem mais complicados na sua estrutura e muito mais poderosos computacionalmente do que uma MT. Mas não se trata de questão, pois o que Turing demonstrou é que *qualquer* algoritmo (programa) executável em qualquer computador, pode ser processado usando uma versão particular de sua máquina, conhecida como Máquina de Turing Universal. Exceto pela velocidade, que é algo dependente do hardware, não há procedimento computacional que qualquer computador possa fazer que não possa ser feito por uma MTU, dados memória e tempo adequados.

O que é uma MTU? Turing a idealizou ao compreender que além dos dados de entrada armazenados na fita, também o próprio 'programa' – as regras do jogo – poderia ser codificado para ser lido como uma entrada pela MTU. Com esta idéia Turing construiu um programa que poderia simular a ação de qualquer programa *P* quando *P* é colocado como parte da entrada. Isto é, Turing elaborou uma MTU.

Como funciona isto? Suponha que se tenha um programa *P* para uma máquina de Turing. O único necessário agora é escrever este programa na fita da MTU, junto com os dados de entrada sobre os quais atuará o programa *P*. Em seguida a MTU irá simular a ação

de *P* sobre os dados. Isto quer dizer também que durante o processamento não haverá diferença entre o que seria rodar o programa *P* na sua máquina original ou o atual processamento da MTU simulando *P*.

Anexo - Astrolábio

Figura 41: Astrolábio

Sua origem data de 200 a.C. na Grécia Clássica, e existem referências de que Apollonius estudou os princípios da projeção estereográfica (método de representação espacial usado no astrolábio). Contudo, quem mais influiu na teoria da projeção espacial foi Hipparchus, nascido em Nicéia, na Ásia Menor (agora Iznik na Turquia), aproximadamente 180 a.C. Ele, que teve grande influência no desenvolvimento da trigonometria, redefiniu e formalizou a projeção como um método para resolver problemas astronômicos.

As primeiras evidências do uso da projeção estereográfica em uma máquina está na escritura do autor romano e arquiteto, Vitruvius (88 - 26 a.C.), o qual, na obra *De architectura,* descreve um relógio (provavelmente de água) feito por Ctesibius em Alexandria. Aparentemente, o relógio de Ctesibius tinha um campo giratório de estrelas atrás de uma armação de arame que indicava a hora do dia. A armação de arame foi construída com base na projeção estereográfica. Há suspeitas de que o primeiro astrolábio tenha sido construído por Claudius Ptolomeu (150 d.C.), pois em diversas partes de seus escritos deixa a impressão de que possuía um instrumento com as características de um astrolábio. Ptolomeu escreveu extensivamente sobre a projeção estereográfica em seu trabalho conhecido como *Planisferium*. Contudo a primeira descrição de um astrolábio é datada do século VI e foi feita por John Filoponos.

Durante muito tempo o uso do astrolábio ficou restrito aos povos persas e islâmicos. No século XI, foi introduzido na Europa através da Espanha, e no século XIII já se encontrava popularizado. O astrolábio só caiu em desuso a partir do século XVII, devido à popularização de instrumentos como o relógio e o telescópio.

Funcionamento do Astrolábio

O astrolábio baseia-se no princípio da projeção estereográfica. Trata-se de um método que permite traçar o mapa do céu em um plano, sem perder suas informações tridimensionais, processo análogo à criação de um mapa da Terra. Esta projeção é

acompanhada de várias linhas e eixos (ver a 2ª figura) de referência que determinam a direção (azimuth) e a altitude das estrelas (altitude, ângulo que fazem com o horizonte), o ângulo de visão do observador (horizonte) e posição em que este se encontra (Zenith).

Partes do Astrolábio

Os astrolábios mais recentes (ver a 1ª figura abaixo) possuem a parte mãe (mater) na qual estão marcadas informações temporais, zodiacais e espaciais, esta parte serve de suporte para todas as outras peças e funciona como um mostrador de cálculo. Após vários ajustes, ponteiros exibem informações computadas pelo astrolábio. Encaixados na face superior da parte mãe existem vários pratos (ver a 2ª figura abaixo) que trazem a projeção estereográfica do céu para determinada latitude durante o dia ou noite. Alguns astrolábios possuem vários pratos que podem ser trocados de acordo com a latitude em que o observador se encontra. Foram fabricados astrolábios com mecanismos que permitiam o ajuste da latitude, mas estes não se tornaram populares devido ao seu custo e complexidade de uso.

Acima destes pratos estereográficos está disposto um componente chamado *rete* (ver a 3ª figura baixo), que permite o ajuste do astrolábio ao movimento da Terra, através de setas que apontam para estrelas de referências. Também encontramos no rete a projeção do caminho do sol.

Atrás do astrolábio há uma régua, utilizada para ver a altitude do objeto celeste. O astrolábio deve ser suspenso perpendicularmente ao solo, e a régua posta na direção do objeto: uma escala exibe o ângulo deste com o solo. Muitos astrolábios foram feitos com escalas trigonométricas para auxiliar os cálculos astronômicos.

Ajuste do Astrolábio

Com o astrolábio na vertical, ajusta-se a latitude de uma das estrelas de referência do rete com a régua, localizada na parte de traz do astrolábio. Então se dispõe o astrolábio na horizontal, com o rete para cima, e orienta-se a seta correspondente à estrela de referência em sua direção. Desta forma, obtêm-se o mapa estereográfico do céu no momento. Alguns ponteiros marcam a hora e a data correspondentes ao ajuste. O processo inverso também pode ser efetuado, conhecendo-se a data e a hora configura-se o astrolábio para obter o mapa do céu.

Usos do Astrolábio

No século 10, Abd Al-Rahmân B. Umar Al-Sufi escreveu um tratado no qual descrevia 1000 usos para o astrolábio. A partir desta informação pode-se ter idéia da flexibilidade que fornece este instrumento. Muitos problemas que requerem matemática sofisticada podem ser resolvidos apenas conhecendo-se o seu funcionamento. Dentre os usos mais simples, pode-se citar: determinação da hora; localização de corpos celestes; cálculo da duração do dia; etc.

É interessante lembrar que, durante a Idade Média, a Astrologia tinha grande influência no cotidiano das pessoas e por este motivo a grande maioria dos astrolábios tinha funções

ligadas ao zodíaco, para facilitar a criação de horóscopos. Entre os Islâmicos o astrolábio era muito utilizado para determinar o horário das orações, e a direção de Meca (as orações Islâmicas são feitas nessa direção), sendo que algumas variações incluíam funções e réguas que facilitavam estas determinações. Por exemplo, para se determinar a hora atual:

1. A altitude do Sol ou de uma estrela é determinada utilizando-se a régua da parte de trás do instrumento.

2. A posição do Sol na elipse é achada fixando a régua angular móvel na data do dia e lendo a longitude do Sol na escala do zodíaco.

3. Na frente do astrolábio, a régua é girada até que cruze a elipse na longitude atual do Sol. O ponto onde a régua cruza a elipse é a posição atual do Sol.

4. São girados o rete e a régua até que o Sol ou ponteiro da estrela esteja na altitude medida.

5. A régua aponta para a hora solar aparente no membro. Tempo solar aparente é o tempo como mostrado em um relógio de sol e é diferente para cada longitude.

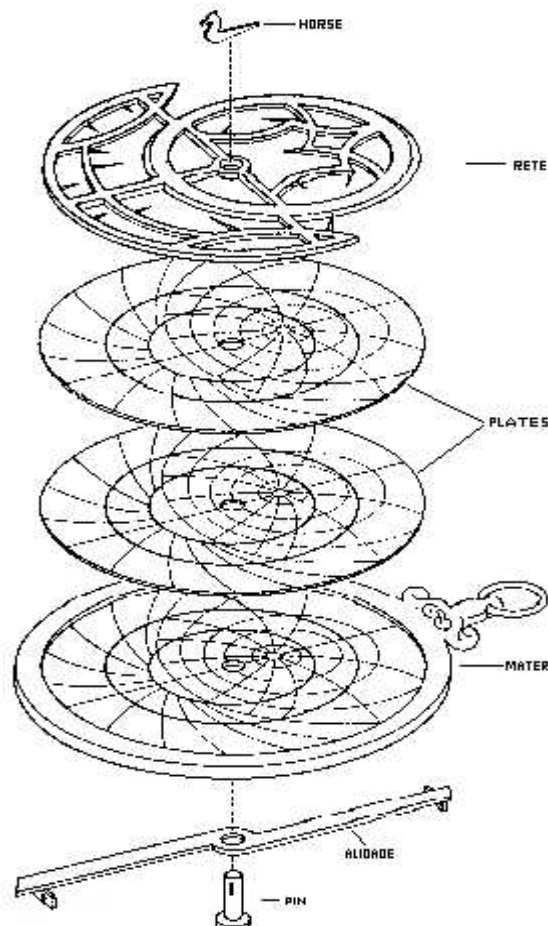



Figura 42: Astrolábio - 1

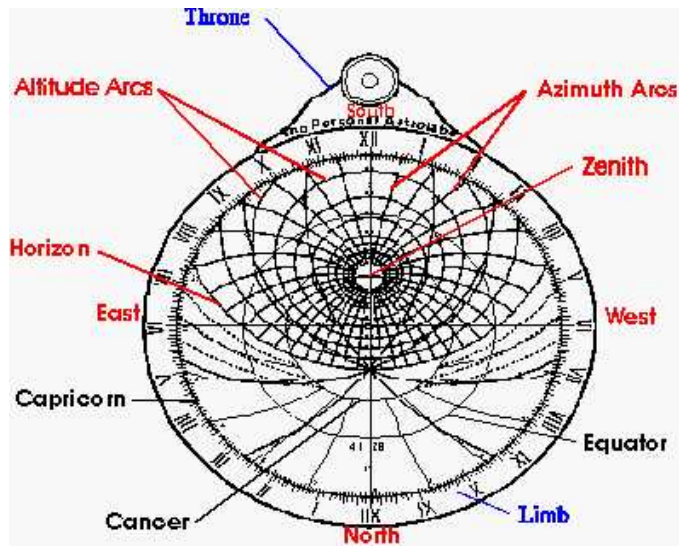



Figura 43: Astrolábio - 2

Figura 44: Astrolábio - 3

Anexo - Turing e a Máquina Enigma*

Antes da guerra os alemães tinham dispendido um esforço considerável no desenvolvimento de um melhor sistema de codificação, e isto era motivo de grande preocupação para o Serviço Secreto Britânico, que no passado conseguira decifrar as mensagens do inimigo com relativa facilidade. A história oficial da guerra publicada, pelo HMSO (Serviço Britânico de Informações na Segunda Guerra Mundial), descreve a situação na década de 1930:

> Por volta de 1937 ficou estabelecido que, ao contrário dos japoneses e italianos, o exército alemão, sua marinha e provavelmente a sua força aérea, junto com organizações estatais, como as das ferrovias e a SS, usavam para tudo, exceto nas comunicações táticas, diferentes versões do mesmo sistema cifrado. A máquina Enigma tinha sido colocada no mercado na década de 1920, mas os alemães a tinham tornado mais segura através de modificações progressivas. Em 1937, a Escola de Cifras e Códigos do Governo tinha quebrado o código do modelo menos modificado e seguro dessa máquina, que estava sendo usado pelos alemães, italianos e pelas forças nacionalistas espanholas. Mas, fora isto, a Enigma ainda resistia ao ataque e parecia que ia continuar assim.

A máquina Enigma consistia de um teclado, ligado a uma unidade codificadora. O codificador tinha três rotores separados e as posições dos rotores determinavam como cada letra no teclado seria codificada. O que tornava o código da Enigma tão difícil de quebrar era o enorme número de modos nos quais a máquina podia ser regulada. Em primeiro lugar, os três rotores na máquina eram escolhidos de uma seleção de cinco que podia ser mudada e trocada para confundir os adversários. Em segundo lugar, cada rotor podia ser posicionado em 26 modos diferentes. Isto significava que a máquina podia ser regulada em milhões de modos diferentes. E além das permutações permitidas pelos rotores, as conexões no quadro de chaveamento, na parte detrás da máquina, podiam ser mudadas manualmente para fornecer um total de 150 trilhões de regulagens possíveis. E para aumentar ainda mais a segurança, os três rotores mudavam de orientação continuamente, de modo que, cada vez que uma letra era transmitida, a regulagem da máquina, e portanto o código, iria mudar de uma letra para outra. Assim, se alguém batesse "DODO" no teclado iria gerar a mensagem "FGTB": o "D" e o "0" eram transmitidos duas vezes, mas codificados de modo diferente a cada vez.

As máquinas Enigma foram fornecidas ao Exército, Marinha e Força Aérea da Alemanha, e eram até mesmo operadas pelas ferrovias e outros departamentos do governo. Mas, como acontecia com os sistemas de código usados naquela época, a fraqueza da Enigma consistia em que o receptor tinha que conhecer a regulagem da máquina que emitira a

* Conforme **[Sin99]**

mensagem. Para manter a segurança os ajustes da Enigma eram mudados diariamente. Um dos meios que os transmissores de mensagens tinham de mudar a regulagem diariamente, enquanto mantinham os receptores informados, era publicar as regulagens num livro secreto de códigos. O risco desta abordagem é que os britânicos podiam capturar um submarino e obter o livro-código com os ajustes diários da máquina para o mês seguinte. A abordagem alternativa, que foi adotada durante a maior parte da guerra, consistia em transmitir a regulagem do dia no princípio da mensagem principal, usando o código do dia anterior.

Figura 45: A máquina Enigma

A criptografia é uma batalha intelectual entre o criador do código e aquele que tenta decifrá-lo. O desafio para o codificador é misturar a mensagem até um ponto em que ela seja indecifrável se for interceptada pelo inimigo. Contudo, existe um limite na quantidade possível de manipulação matemática devido à necessidade de enviar as mensagens de modo rápido e eficiente. A força do código alemão da Enigma consistia em que a mensagem passava por vários níveis de codificação a uma velocidade muito alta. O desafio para o decifrador do código era pegar uma mensagem interceptada e quebrar o código quanto o conteúdo da mensagem ainda fosse relevante.

Turing liderou uma equipe de matemáticos que tentou construir réplicas da máquina Enigma. Ele incorporou nesses engenhos as suas idéias abstratas, anteriores à guerra. A idéia era verificar todos os ajustes possíveis da Enigma até que o código fosse descoberto. As

máquinas britânicas tinham dois metros de altura e eram igualmente largas, empregando relês eletromecânicos para verificar todos os ajustes possíveis da Enigma. O constante tiquetaquear das máquinas deu-lhes o apelido de bombas. Apesar da sua velocidade, era impossível que as bombas verificassem cada uns dos 150 trilhões de ajustes possíveis da Enigma dentro de um tempo razoável. Por isso a equipe de Turing teve de procurar meios de reduzir significativamente o número de permutações extraindo toda a informação que pudesse das mensagens enviadas.

Um dos grandes saltos em direção ao sucesso aconteceu quando os britânicos perceberam que a máquina Enigma não podia codificar uma letra nela mesma. Ou seja, se o emissor teclasse “R”, então, dependendo do ajuste, a máquina poderia transmitir todo tipo de letra, menos “R”. Este fato, aparentemente inócuo, era tudo o que necessitavam para reduzir drasticamente o tempo necessário para decifrar as mensagens. Os alemães contra-atacaram limitando o comprimento das mensagens que enviavam. Todas as mensagens, inevitavelmente, continham indícios para a equipe de decifradores do código, e quanto maior a mensagem, mais pistas ela continha. Ao limitar as mensagens a um máximo de 250 letras, os alemães esperavam compensar a relutância da Enigma em codificar uma letra como a mesma.

Com o fim de quebrar os códigos, Turing freqüentemente tentava adivinhar palavras chaves nas mensagens. Caso acertasse, isto aceleraria enormemente a decodificação do resto da mensagem. Por exemplo, se os decodificadores suspeitavam de que uma mensagem continha um relatório meteorológico (um tipo freqüente de relatório codificado), então eles supunham que a mensagem conteria palavras como “neblina” ou “velocidade do vento”. Se estivessem certos, podiam decifrar rapidamente a mensagem e, portanto, deduzir o ajuste da Enigma para aquele dia em particular. E pelo resto do dia outras mensagens, mais valiosas, seriam decifradas facilmente. Quando fracassavam na adivinhação de palavras ligadas ao tempo, os britânicos tentavam se colocar na posição dos operadores alemães da Enigma para deduzir outras palavras chaves. Um operador descuidado poderia chamar o receptor pelo primeiro nome ou ele poderia desenvolver idiossincrasias conhecidas pelos decifradores. Quando tudo o mais falhava e o tráfego alemão de mensagens fluía sem ser decifrado, a Escola Britânica de Códigos podia até mesmo, dizem, recorrer ao recurso extremo de pedir à RAF (Força Aérea Britânica) para que minasse um determinado porto alemão. Imediatamente o supervisar do porto atacado iria enviar uma mensagem codificada que seria interceptada pelos britânicos. Os decodificadores teriam certeza então de que a mensagem conteria palavras como "mina", “evite" e "mapa de referências". Tendo decodificado esta mensagem, Turing teria os ajustes da Enigma para aquele dia e quaisquer mensagens posteriores seriam vulneráveis à rápida decodificação.

No dia 1º de fevereiro de 1942 os alemães acrescentaram uma quarta roda às máquinas Enigma usadas para enviar mensagens particularmente importantes. Esta foi a maior escalada no nível de codificação durante a guerra, mas finalmente a equipe de Turing respondeu aumentando a eficiência das bombas. Graças à Escola de Códigos, os aliados sabiam mais sobre seu inimigo do que os alemães poderiam suspeitar. O impacto da ação dos submarinos no Atlântico foi grandemente reduzido e os britânicos tinham um aviso prévio dos ataques da Luftwaffe. Os decodificadores também interceptavam e decifravam a posição

exata dos navios de suprimentos alemães, permitindo que os destróiers britânicos os encontrassem e afundassem.

Mas o tempo todo as forças aliadas tinham que ter cuidado para que suas ações evasivas e ataques precisos não revelassem sua habilidade de decodificar as comunicações alemãs. Se os alemães suspeitassem de que o código da Enigma fora quebrado, eles iriam aumentar seu nível de codificação mandando os britânicos para a estaca zero. Por isso houve ocasiões em que a Escola de Códigos informou aos aliados sobre um ataque iminente e o comando preferiu não tomar medidas extremas de defesa. Existem mesmo boatos de que Churchill sabia que Coventry seria o alvo de um bombardeio devastador mas preferiu não tomar precauções especiais para evitar que os alemães suspeitassem. Stuart Milner Barry, que trabalhou com Turing, nega o boato. Ele diz que a mensagem relevante sobre Coventry só foi decifrada quando já era tarde demais. Este uso contido da informação codificada funcionou perfeitamente. Mesmo quando os britânicos usavam as mensagens interceptadas para causar perdas pesadas no inimigo, os alemães não suspeitaram de que o código Enigma fora quebrado. Eles pensavam que seu nível de codificação era tão alto que seria totalmente impossível quebrar os códigos. As perdas excepcionais eram atribuídas à ação de agentes britânicos infiltrados.

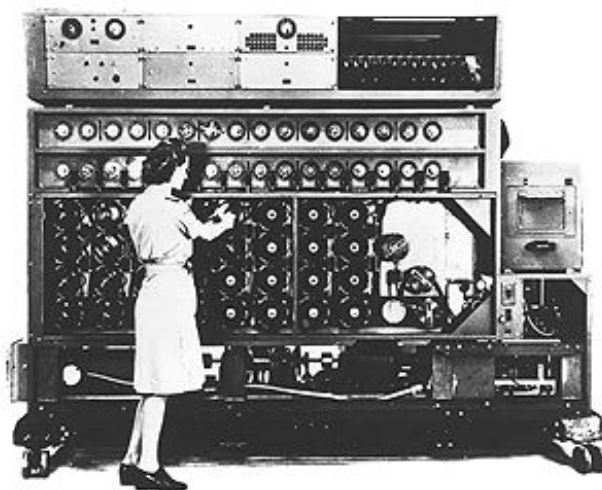



Figura 46: Uma *bombe*, máquina inicialmente usada para tentar decifrar código

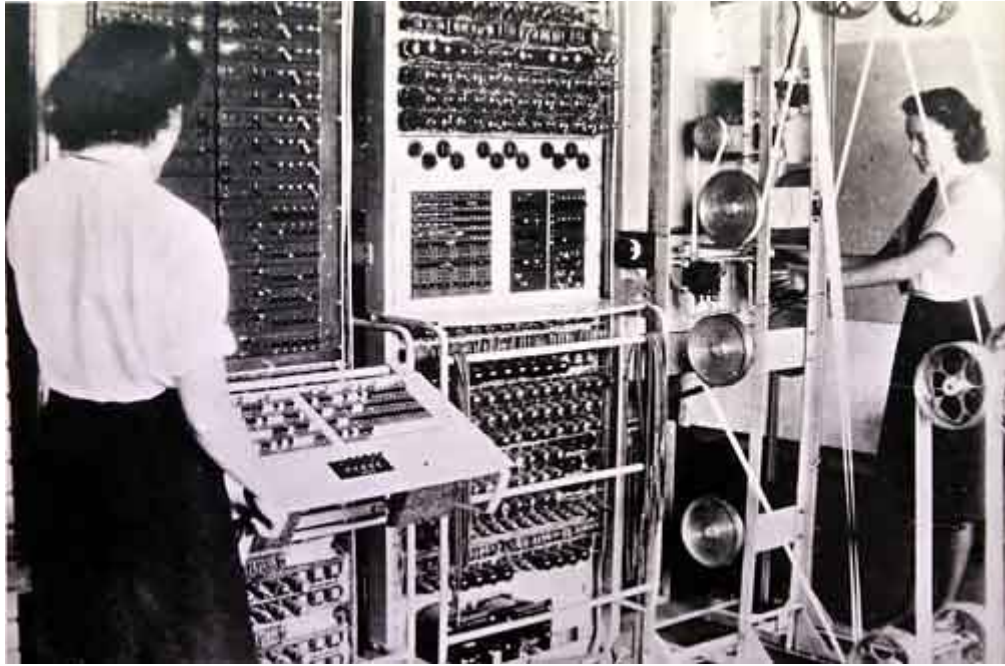



Figura 47: Computador COLOSSUS, que substituiu as *bombes*

Índice de Figuras